ORSAY
Série A. N°
N° D'ordre :
132

THESE

présentée

A LA FACULTE DES SCIENCES

D' ORSAY

UNIVERSITE DE PARIS

Pour obtenir le titre de Docteur-Ingénieur

SPECIALITE : ELECTRONIQUE (TRAITEMENT DE L'INFORMATION)

par

Claude BARBE

Sujet de la Thèse : Système d'Exploitation en Temps Partagé - Etude d'un Système de Pagination programmé

Soutenue le 2 Avril 1971 devant la Commission d'Examen

MM. PICINBONO Président

DEBRAINE
DONADIEU Examinateurs

T H E S E

présentée

A L A F A C U L T E D E S S C I E N C E S

D' O R S A Y

UNIVERSITE DE PARIS

Pour obtenir le titre de Docteur-Ingénieur

SPECIALITE : ELECTRONIQUE (TRAITEMENT DE L'INFORMATION)

par

Claude BARBE

Sujet de la Thèse : Système d'Exploitation en Temps Partagé - Etude d'un Système de Pagination programmé

Soutenue le 2 Avril 1971 devant la Commission d'Examen

MM. PICINBONO Président

DEBRAINE
DONADIEU Examinateurs

UNIV.
LEIDEN
BIBL.

Au terme de ce rapport, je tiens à remercier en Monsieur le Professeur GREMY tous les membres du Centre de Calcul et de Statistiques des Facultés de Médecine de Paris pour l'accueil qui m'y a été réservé et pour les conditions et moyens de travail mis à ma disposition sans aucune restriction.

Je tiens à remercier par dessus tout Monsieur DEBRAINE pour les conseils et les conditions matérielles fournies sans lesquels ce travail n'aurait sans doute pas vu le jour.

Je n'aurai garde d'oublier la diligence et le soin avec lesquels Madame AIGUESPARSES a assuré la frappe de ce texte malgré des délais rigoureux.

C'est enfin à Monique DUHAMEL que vont mes remerciements pour ses encouragements et sa patience à me relire et à me suivre.

Ce travail est le résultat de 18 mois passés au sein de l'équipe système du Centre de Calcul des Facultés de Médecine de Paris, situé au Centre Hospitalier Universitaire Pitié-Salpétrière.

Au cours de ce séjour, il m'a été permis :

- de participer à la maintenance du système d'exploitation et des programmes de soutien en étudiant leurs défaillances occasionnelles,

- de participer à la mise au point d'une liaison à grande vitesse avec un ordinateur éloigné servant de concentrateur de consoles,

- d'étudier le matériel et quelquefois de le réparer.

L'ordinateur dont il sera question et qui équipe ce centre est un P.D.P. 10 fabriqué par Digital Equipment Co. (Maynard - Massachusetts - USA). Il possède une mémoire de 64 K mots de 36. bits (près de 300 K octets) en ferrite rapide (lecture - écriture en 1 microseconde), un disque rapide à têtes fixes et plus de 5 millions de mots sur disque amovible.

Le système d'exploitation fourni par le constructeur est exclusivement du type "temps partagé" pour 1 à 63. travaux simultanés. Il s'agit du système 50 gérant la mémoire virtuelle et la réentrance, la version actuelle (septembre 70) 4-72 gère les fichiers sur les 2 types de disques.

Les langages disponibles sont l'assembleur (MACRO 10), le FORTRAN, le BASIC, l'AID (Algebraic Interpretive Dialogue) et le COBOL.

Pour ce travail, la documentation utilisée a été celle de la machine, les manuels de référence et de maintenance et les listes du superviseur.

SOMMAIRE

pages

1 - PRESENTATION DU PARTAGE DE TEMPS 1

2 - LE MATERIEL 5

I - Structure possible d'une machine travaillant en temps réel 5

1 - Organisation générale 5

2 - L'unité centrale 9

3 - L'horloge 12

II - Caractéristiques indispensables au partage de temps 13

1 - Le registre de base et le registre de protection 13

2 - Les seconds registres et la réentrance 15

3 - LE SUPERVISEUR 17

I - Démarrage du système - initialisations 18

II - La gestion des consoles 20

III - L'interruption d'horloge et les tâches différées 23

1 - Le report en priorité basse 23

2 - L'interblocage avec les tâches en mode exécutif 26

3 - Le décodage et l'exécution des ordres 28

4 - Exemple d'ordre : charger un programme et le lancer 31

IV - Les partages 34

V - La gestion de la mémoire 35

1 - Mémoire réelle 35

2 - Mémoire virtuelle 35

3 - Réservation de mémoire - routine générale 36

4 - Réservation de mémoire par opérateur programmé 40

5 - Réservation de mémoire par ordre 42

VI - Les échanges entre mémoire réelle et mémoire virtuelle 43

1 - Organisation de la mémoire virtuelle 43

2 - Amorçage du processus d'échange 45

3 - Le système de tassement en mémoire réelle 46

4 - Sortie d'un segment : conditions à vérifier au préalable 47

pages

VII - Gestion des programmes partagés 48
1 - Segment pur et segment impur 48
2 - La protection écriture du segment haut 49
3 - Effet sur les sorties en mémoire virtuelle 49
4 - Création d'un segment réentrant 50
5 - Appel d'un tel segment, compatibilité avec les programmes en un seul segment, possibilité d'inexistence d'un segment bas associé dans le système 50
6 - La protection écriture et les conditions de sa levée 53
7 - Problèmes du maintien inactif en mémoire des segments réentrants : avantage de leur survie 54

VIII - Gestion des tours de passage 56
1 - L'attribution du numéro de travail, liaison avec la console de l'utilisateur 56
2 - Les files d'attente du système 57
3 - Déclenchements des changements de queue 59
4 - Queues d'attente des périphériques partagés 61
5 - Les 3 queues de passage, priorités et évolution 62
6 - Algorithme de recherche des travaux à sortir de la mémoire réelle ou à rentrer en mémoire réelle 65

IX - Le décodeur des opérateurs programmés 66
1 - Définition et signification des opérateurs programmés 66
2 - Opérateurs formalisés d'entrée-sortie 67
3 - Appellation symbolique des périphériques physiques 69
4 - Organisation du décodage pour les entrées-sorties 71
- routine générale pour les canaux programmés
- les tables des programmes et les tables d'état-d'équipement-physique
5 - Opérateur programmé d'initialisation 75
6 - Demande de construction de tampons, structure d'une série de tampons, accès séquentiel aux caractères d'un fichier 76

pages

7 - Opérateurs programmés en lecture ou écriture d'un fichier 79
8 - Opérateur programmé d'entrée 81
a) - remplissage asynchrone des tampons 82
b) - routines prioritaires 85
c) - problèmes de synchronisation, remplissage - utilisation 89
d) - utilisation des contrôleurs de périphériques par "bouffées" 90
9 - Opérateur programmé de sortie 91
10 - Fermeture de fichier en entrée 93
11 - Fermeture de fichier en sortie 94
12 - Libération d'équipement 95

X - Utilisation d'un équipement partageable multiplement accessible (disques, tambours) 98
1 - Organisation logique 99
2 - Nature et hiérarchie des informations stockées 100
a) - blocs d'identification 100
b) - fichiers répertoires 102
c) - fichier répertoire général 104
3 - Initialisation logique de l'équipement DISQUE 106
4 - Ouverture d'un fichier en lecture partageable 107
5 - Ouverture d'un fichier en écriture 108
. détermination de l'état d'un fichier dans le système
6 - Entrée du tampon suivant (routine spécifique d'entrée) 114
7 - Sortie du tampon suivant (routine spécifique d'entrée) 116
8 - Procédure d'interruption prioritaire : cas du disque 118
9 - Chaînage logique des ordres de transferts en attente pour les usagers du système 121
10 - Fermeture d'un fichier ouvert en lecture 123
11 - Fermeture d'un fichier ouvert en écriture 126

pages

12 - Libération de l'équipement DISQUE 129
13 - Modification de l'identification d'un fichier 131

4 - ETUDE D'UN SYSTEME DE PAGINATION PROGRAMME DANS UNE STRUCTURE DE PARTAGE DE TEMPS 133
1 - Présentation du problème 133
2 - Aspect du programme avant exécution 135
3 - Contrôle du programme paginé : moniteur 135
4 - Causes d'interruption de programme 135
a) - adresse hors page 136
b) - opérateur programmé 137
5 - Implantation du programme (1 page) et du moniteur 137
a) - définition d'une page 138
b) - propriété de translatabilité, utilisation du second système de translation - protection 140
6 - Position du moniteur de pagination 142
a) - conditions imposées par le superviseur 142
b) - niveau usager de page 143
c) - changement de mode (normal - page) 145
d) - modifications du superviseur 146
7 - Ecriture des programmes paginés 147
8 - Organisation du moniteur de pagination 151
9 - Interruption pour dépassement de la protection 154
a) - différents cas 155
b) - ambiguïtés rencontrées et solutions proposées 157
10 - Interruptions par les opérateurs programmés ou les instructions illégales 163
11 - Conclusions sur ce système de pagination programmé 167

5 - ANNEXES 169
A - Le matériel utilisé 169
I - Les codes d'instruction 169
1 - Choix d'une taille de mot machine 169
2 - Classification des instructions 170
3 - Revue des instructions de base 172

pages

II - Principes de fonctionnement de l'unité centrale 179
1 - Registres-machine et connexions 179
2 - Fonctionnement asynchrone 180
3 - Mécanisme d'interruption 182
4 - Schéma des séquences de l'exécution d'une instruction 184

B - Construction d'un superviseur sur mesure 187

C - Correspondance entre les expressions françaises et les expressions américaines originales 190

D - Listes du moniteur 196

CHAPITRE I

PRESENTATION DU PARTAGE DE TEMPS

Dans un ensemble complexe de traitement de l'information comme une installation en partage de temps, il y a différents niveaux qui contribuent au bon fonctionnement du tout, ce sont par importance décroissante :

- le matériel
  - l'unité centrale
  - la mémoire centrale
  - les unités d'échange
  - les mémoires auxiliaires
    - . accès sélectif
    - . accès séquentiel

- le programme superviseur gestionnaire de ce matériel

- le programme superviseur gestionnaire des programmes particuliers

- les programmes généraux de la bibliothèque
  - au service du superviseur
  - au service de la maintenance
  - au service de l'usager

- les programmes d'application propres à chaque usager de l'ensemble.

La réunion de ces moyens de traitement de l'information qui agissent solidairement pour manipuler des données en vue d'exécuter les ordres d'un utilisateur (de ces moyens), constitue le système. Les programmes particuliers d'application n'en font pas partie, ils sont la propriété de l'usager qui les a créés. L'usager s'il apparaît toujours ici comme le cas particulier au bas de l'échelle dans l'importance

relative des moyens rassemblés, constitue en fait la raison d'être de l'ensemble et sa satisfaction la finalité du système tout entier. Cette importance relative du système et de l'usager est celle de l'intérêt général sur le particulier, importance que nous saisissons ici en partant des éléments les mieux définis et les plus sûrs que sont les circuits et en montant dans la diversité que forment les programmes-objets des divers usagers possibles. Le système tel que nous l'envisageons doit chercher à offrir à un usager un service garanti à la fois adapté à ses problèmes et muni d'une généralité telle que ce même service soit tout aussi adapté à une classe de problèmes apparemment différents chez un autre usager. Cette adaptation est surtout effectuée par les programmes généraux de la bibliothèque au service de l'usager. Ces programmes ne sont en fait que des programmes habilement construits pour de multiples applications potentielles mais qui ne se distinguent en rien quant à leur composition des éléments (instructions, fichiers, etc...) mis à la disposition de tous les usagers. Ces programmes généraux ce sont les traducteurs de langages plus ou moins évolués, plus ou moins abstraits, en ordres beaucoup plus simples destinés au matériel et au superviseur : on peut citer les compilateurs, les assembleurs, les créateurs et modificateurs de fichiers, les transporteurs d'information.

Le système doit offrir un service garanti. Il est en effet essentiel qu'un usager du système puisse faire l'économie de l'étude qui va suivre et qu'il puisse aborder le système en n'en ayant qu'une connaissance extérieure et globale. Cette connaissance globale qui fait abstraction des phénomènes plus microscopiques est heureusement suffisante dans les cas les plus usuels, c'est grâce à elle que les problèmes complexes peuvent être résolus. Cette connaissance globale n'est profitable que si les éléments appréhendés sont simples et sûrs et ce n'est que lorsque la confiance est mise en défaut qu'un approfondissement de la connaissance est indispensable pour trouver des éléments plus microscopiques suffisamment fiables à leur tour. Pour

être d'un usage le plus général possible avec le moins d'efforts un système informatique doit être d'une grande sûreté et faire abstraction des phénomènes physiques réels qui le rendent possible, sa personnalité doit être intégre, facile et néanmoins nuancée. Les problèmes généraux doivent pouvoir être abordés et résolus dans des langages généraux (par exemple ordre de recopie de fichier) tandis que des problèmes un peu plus particuliers peuvent également être résolus, formulés dans des langages disponibles plus compliqués mais plus proches aussi des mécanismes physiques. Cette analyse et ce rapprochement des phénomènes plus élémentaires qui participent à l'exécution d'un ordre d'un usager, s'arrêtent pour cet usager du système au langage d'assemblage, au langage machine et aux instructions programmées dont le mécanisme lui échappe bien que les effets en soient rigoureusement décrits dans les manuels mis à sa disposition. Un usager d'un système ignore comment une instruction est décodée, il ignore le temps qui séparera deux instructions consécutives même s'il connaît la durée de chacune individuellement. Pour en savoir davantage, il faut analyser les parties vitales du système lui-même : le matériel et le superviseur. C'est l'objet de cette étude. Un système aussi évolué, aussi complet qu'il est possible et souhaitable est évidemment complexe et par conséquent coûteux à mettre au point et à entretenir. Si un système a été créé d'abord très général, ce fut pour des raisons de débouchés, pour élargir ses possibilités de vente sans adaptation coûteuse, mais cette généralité est aussi un atout indispensable à l'échelon d'une seule installation si cette installation est utilisée conjointement pour résoudre des problèmes très diversifiés. Dans ces conditions une seule configuration de système peut suffire à satisfaire les différents besoins locaux. Ces besoins qui furent d'abord des besoins de puissance, de calcul et de grosses manipulations d'information sont devenus aussi besoins d'une réponse très rapide et de développement de nouveaux programmes dont l'écriture et la mise au point nécessita une assistance accrue de la machine afin de suppléer au manque de personnel qualifié et d'accélérer les réalisations.

Cette mise au service de l'individu, de systèmes puissants au propre rythme de cet individu (système conversationnel) n'était économiquement possible que par le partage des ressources rassemblées pour constituer chaque système entre beaucoup d'individus servis quasi simultanément par un système infiniment plus rapide que chacun individuellement et suffisamment agile pour n'en léser aucun tout en utilisant globalement au maximum les ressources communes.

Lorsqu'un système a su se répartir lui-même entre plusieurs usagers le système en temps partagé était né. C'est un tel système qui va être étudié ici.

Plus que dans tout autre système, dans une installation en partage de temps, l'aspect extérieur saisi par un utilisateur diffère grandement de la réalité par l'impression d'être seul à utiliser le système. Si le système est suffisamment sûr, on peut alors raisonnablement définir pour cet utilisateur une machine virtuelle ayant ses propres langages et des propriétés caractéristiques de l'emprise de type échantillonné de cet utilisateur sur le système entier. Les performances de temps, si elles dépendent de la charge totale du système à un instant donné peuvent pour des tâches très simples être toujours suffisamment bonnes pour que l'utilisateur ne s'aperçoive pas de la différence et se croie toujours obéi en maître unique des moyens mis à sa disposition, cette apparence étant due à l'incapacité d'un utilisateur de discerner un temps de réponse dans un rapport de 1 à 100 quand son oeil ou sa machine à écrire ont des temps de réponse encore bien plus élevés. Domaine des besoins modestes mais multiples le système en temps partagé devra veiller à toujours garantir un temps de réponse court quitte à faire attendre tout naturellement des travaux plus lourds dont ce n'est pas le mode de traitement idéal.

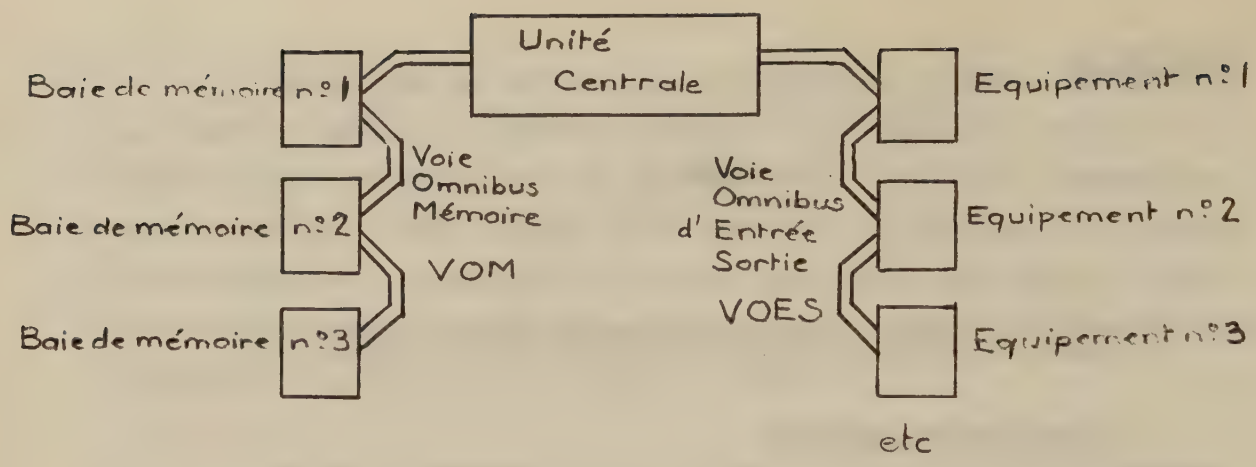
CHAPITRE 2

LE MATERIEL

I - STRUCTURE POSSIBLE D'UNE MACHINE TRAVAILLANT EN TEMPS REEL

1 - Organisation générale

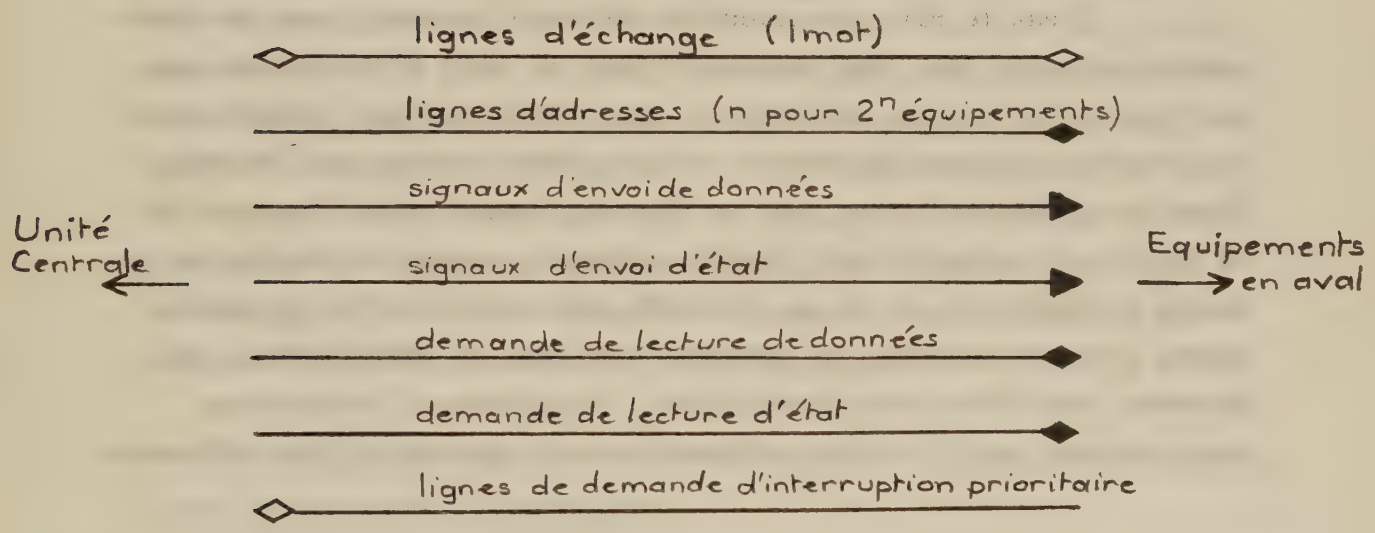
Une machine destinée à un système d'exploitation en partage de temps nécessite d'abord les attributs d'une machine destinée à travailler en temps réel. Une structure modulaire est souhaitable pour permettre une configuration selon les besoins de chaque installation et une extension future au moindre coût. Une solution possible est de disposer d'une unité centrale capable de réaliser elle-même toutes les fonctions d'entrée-sortie moyennant des règles très simples de branchement d'équipements périphériques, ceux-ci étant alors faciles à construire, à adapter et à gérer par programmation. Cette solution paraît souhaitable quand chaque installation risque d'être amenée à connecter à l'organe de calcul des équipements d'échange d'information originaux. Ceci suppose un organe unique, l'unité centrale, maître de l'ensemble, facilement disponible pour satisfaire les besoins des périphériques travaillant à leur rythme propre et facilement raccordable à un nouvel équipement lors des modifications de configuration. A l'image des structures de listes, c'est la voie omnibus d'échange d'information qui offre le moyen le plus simple de satisfaire de telles exigences. La seule limite au nombre de stations sur une telle voie est la capacité d'adressage adoptée à la construction de l'unité centrale. Compte tenu du rythme des échanges et des fonctions demandées une structure à 2 voies l'une vers la mémoire rapide, l'autre vers les équipements d'entrée-sortie a été adoptée dans la machine étudiée.



Cette structure suppose simplement que chaque équipement et chaque banc de mémoire dispose du moyen de reconnaître sa propre adresse envoyée sur la voie qui le traverse et d'y répondre de façon appropriée. Ce qui distingue les mémoires, c'est leur passivité apparente; en fait s'il appartient en même temps à une autre voie omnibus, un banc de mémoire déjà accupé sur une de ses portes d'accès pourra éventuellement faire attendre sur l'autre porte l'unité centrale qui l'interroge. Les équipements d'entrée-sortie ont, eux, une évolution qui leur est propre et ont besoin, à la réalisation de certaines conditions, que des décisions soient prises en fonction d'un contexte qui leur échappe et que seule l'unité centrale pilotée par des programmes rangés en mémoire rapide est capable de connaître. Ainsi un caractère issu d'une machine à écrire a été reçu en série, rangé en parallèle dans un registre de l'équipement de contrôle de ces machines à écrire, sa parité vérifiée. Ce caractère doit être rangé ailleurs, retrouvable de surcroît, avant qu'un nouveau n'arrive, frappé par l'utilisateur impatient. C'est pourquoi toutes les machines travaillant en temps réel disposent d'un système d'interruption de l'unité centrale, pour libérer celle-ci de tout rôle d'attente ou de surveillance des multiples équipements externes. Pendant que l'utilisateur tapait sur sa touche, l'unité centrale dégagée de tout souci faisait des opérations utiles. On multiplie

les lignes d'interruption pour établir un ordre dans les interruptions possibles venant simultanément d'équipements différents. Cet ordre se traduit par la priorité qu'une ligne d'interruption peut avoir sur une autre, une interruption plus prioritaire doit pouvoir interrompre un travail entrepris pour le compte d'un signal d'interruption issu d'une ligne moins prioritaire.

Il n'est pas indispensable de disposer de beaucoup de lignes d'interruptions prioritaires si celles-ci peuvent être facilement affectées et désaffectées ou ce qui revient au même, leurs niveaux de priorité relatifs peuvent être changés à volonté, c'est-à-dire par programme pour chaque équipement de la voie omnibus d'entrée-sortie. Les priorités entre les différentes lignes peuvent être facilement câblées définitivement pour faciliter la programmation.



Structure de la voie omnibus d'entrée-sortie

Une demande d'interruption sur une ligne se traduit en pratique par mise en court-circuit de cette ligne par un des équipements, et positionnement d'un ou de plusieurs bits dans les registres d'états lisibles par l'unité centrale dans cet équipement. Plusieurs équipements peuvent sur une même ligne d'interruption déposer une demande, l'unité centrale interrogera par programme tous les équipements susceptibles d'avoir provoqué une telle interruption et agira en changeant l'état de cet équipement pour faire cesser la cause d'interruption, celle-ci se prolongera si une autre demande existait simultanément sur la même ligne jusqu'à ce que toutes les causes aient été neutralisées par les actions spécifiques attendues des différents équipements.

La demande d'interruption par un équipement abandonné à lui-même jusqu'à ce qu'il ait accompli certaine action en général commandée par l'unité centrale, doit être prise en compte avec plus ou moins de promptitude en vue d'assurer un bon fonctionnement de cet équipement (cadence de lecture ou d'écriture d'une bande magnétique en mouvement).

C'est le rôle des priorités câblées d'assurer, pour les équipements affectés dans leur registre d'état du droit d'interrompre sur une ligne plus ou moins prioritaire, une prééminence non ambiguë entre les besoins de chaque équipement. On peut même requérir des interruptions sur plusieurs lignes pour un même équipement selon l'urgence de la fonction à accomplir par l'unité centrale : un mot de données ne pourra attendre plus de 50 µs, un arrêt pour erreur ou fin de fichier pourra attendre davantage et demander bien plus de vérifications programmées, donc être moins prioritaire ; les niveaux d'interruption pour ces deux cas d'un même équipement auront avantage à être différents.

2 - L'unité centrale

Pour ne pas spécialiser l'unité centrale, il suffit que l'adjonction d'un nouvel équipement spécifique à l'installation n'entraîne finalement que l'insertion sur la VOES d'un dispositif de décodage d'adresse et de réponse aux signaux de données et d'état quand l'adresse a été décodée. Sous réserve de respecter les caractéristiques techniques d'adaptation à la VOES commune à tous les équipements, tous les autres registres de l'équipement considéré pourront n'être qu'affaire de compromis entre le coût du matériel et le nombre d'instructions de l'unité centrale nécessaires pour réaliser l'échange d'information entre le programme (en mémoire) et le support extérieur, finalement, de la charge que l'utilisation de cet équipement d'échange apportera au système complet : par exemple, l'équipement rudimentaire peut provoquer une interruption à chaque caractère traité ou bien faire lui-même un assemblage en mot et attendre qu'un mot complet soit assemblé pour provoquer cette même interruption. Avec des équipements aussi simples traitant le caractère ou le mot, il est nécessaire que le transfert à la mémoire soit faisable par l'unité centrale elle-même. C'est évidemment une charge pour elle mais cela dispense d'organe(s) autonome(s) spécialisé(s) du type canal à accès direct à la mémoire sans rendre celui-ci impossible. L'utilisation d'un tel appareillage spécialisé n'est en fait justifié qu'au delà d'un certain débit d'échanges en deça duquel il revient moins cher d'utiliser l'unité centrale pour les transferts de données au moyen d'équipements périphériques plus simples et d'un système d'interruption satisfaisant.

Cet usage général de l'unité centrale requiert des caractéristiques particulières déterminantes :

- il faut qu'elle soit rapide,

- il faut qu'elle admette rapidement les interruptions et ceci d'autant plus qu'elle sera amenée à faire (peu fréquemment sans doute) des

échanges à haut débit, ne souffrant pas d'attente, de par l'inertie du support extérieur contrôlé par un équipement limité en registres : bande magnétique (registre d'un mot pour écriture caractère), carte (registre colonne) et à plus forte raison disque ou tambour magnétique.

La rapidité sera assurée au moyen de transferts en parallèles des mots machine et par une structure asynchrone. Les instructions sont exécutées par des petites routines câblées sans horloge centrale, la mémoire est asynchrone et répond dès qu'elle est disponible. L'interruptibilité facile sera assurée par l'absence d'instruction ininterruptible dans un certain délai, les instructions multiples seront judicieuses : il n'y a pas d'instruction se modifiant elle-même en cours d'exécution mais des instructions multiples à pointeur modifiable permettant une reprise en cas d'interruption.

Les registres généraux et les registres d'index ont intérêt à être d'autant plus rapides d'accès qu'ils sont relativement plus souvent utilisés, donc d'une technologie différente plus rapide et plus chère que le reste de la mémoire. Plutôt que d'avoir des instructions spéciales de manipulation de registres d'index, il est possible d'utiliser les instructions générales de transfert à condition que les registres d'index soient identiques aux registres généraux (leurs parties droites exclusivement) et plutôt que d'avoir des instructions de type registre-registre, il est aussi possible d'utiliser les instructions générales de transfert à condition que les adresses permettent de les désigner sans ambiguïté au même titre que les adresses de mémoire centrale. Ceci offre le double avantage, outre l'unicité des instructions de copie et de rangement, de permettre le choix entre des registres rapides spécialement montés et l'utilisation des positions d'adresses réservées de la mémoire centrale (Ø. à 15.) et aussi de suppléer à une défaillance des registres rapides en les remplaçant par les adresses correspondantes et non utilisées de la mémoire centrale sans rien changer à la programmation.

L'unité centrale est équipée de registres d'états (différents des 16 registres de travail) lisibles par les instructions générales. En un sens l'unité centrale est considérée comme un équipement dont les défaillances doivent être diagnostiquées immédiatement. Ces défaillances peuvent être réelles :

- coupure de l'alimentation du secteur, parité recalculée du mot lu en mémoire centrale ne coïncidant pas avec celle reçue de la mémoire. La mémoire n'a pas répondu à une demande de lecture ou d'écriture.

- elles sont plus souvent dues non aux circuits, mais aux opérandes, c'est-à-dire au programmeur ; l'unité centrale doit avertir tout de suite qu'il y a un résultat anormal : épuisement du pointeur de pile, dépassement de capacité, adresse supérieure à la capacité de la mémoire. Pour parvenir à cette action immédiate, l'équipement "unité centrale" comme les autres équipements d'entrée-sortie peut se voir attribuer par programme un niveau de priorité (une ligne d'interruption) et par des instructions d'entrées-sorties, ses registres peuvent être initialisés et testés de manière à retrouver plus tard la cause de l'interruption.

Avec le mécanisme d'interruption lui-même, ce sont les deux équipements (adresses Ø et 1) imposés dans le système, les 126. autres étant libres d'emploi au gré des installations.

3 - L'horloge

Dans une installation destinée à traiter de l'information en temps réel et à plus forte raison lorsque l'exploitation devra être partagée dans le temps, il est indispensable de disposer d'un moyen d'estimation du temps écoulé entre deux événements. L'unité centrale répétant une même instruction d'incrémentation pourrait bien sûr constituer une horloge mais cela l'immobiliserait très inutilement. Certaines machines disposent d'une horloge indépendante interrogeable par instruction quand on en a besoin ou bien mettant automatiquement à jour une position de mémoire connue ainsi incrémentée indépendamment du programme exécuté. Aucune de ces solutions ne convient tout à fait à la machine étudiée pour laquelle l'économie de registres est une règle et la non-spécialisation des adresses hors des cellules d'interruption un avantage pour l'utilisation et la programmation. Le temps est signalé régulièrement par le "secteur" (20 ms) sous forme d'interruption si cette interruption a été au préalable autorisée, cette interruption se traduit par le positionnement d'un drapeau caractéristique de l'équipement "unité centrale".

Le programme qui s'occupe des interruptions de l'unité centrale tiendra compte s'il le faut de ces signaux réguliers pour mettre à jour une horloge interne au programme en machine sous la forme qui lui plaira.

II - CARACTERISTIQUES INDISPENSABLES AU PARTAGE DE TEMPS

L'horloge de temps réel devient l'organe le plus rigoureux pour équilibrer l'usage du bien commun dans une exploitation en temps partagé. Les signaux extérieurs seront utilisés lors de l'enchaînement des tâches multiples coexistant dans le système.

1 - Le registre de base et le registre de protection

Un besoin déjà ressenti dans les systèmes à un seul utilisateur est la protection de l'utilisateur contre lui-même, notamment lors des mises au point ; il faut pouvoir protéger contre toute destruction malheureuse des parties de programmes déjà mises au point sous peine d'avoir à repartir d'un système indemne pour continuer ; ce qui est un inconvénient sur les petites machines est intolérable sur un système important et incompatible avec des tâches indépendantes censées s'ignorer. Il s'agit de protéger d'abord le programme superviseur contre l'utilisateur maladroit et dans les systèmes importants et d'usage général, de protéger les utilisateurs entre eux contre les accidents d'abord, contre les indiscrétions ensuite. Cette protection doit d'abord s'appliquer au niveau de la mémoire centrale partagée entre les utilisateurs présents. Elle devra s'appliquer comme on le verra aux fichiers conservés sur des supports partagés ou non.

Le mécanisme de protection s'applique à toute la mémoire centrale dans laquelle seule une zone est adressable par l'utilisateur qui peut librement y lire, écrire ou exécuter des instructions. Toute tentative de sortie de cette zone est sanctionnée par une interruption prioritaire traitée par le superviseur en mode privilégié. Les mécanismes utilisés pour réaliser cette protection de la mémoire sont plus ou moins compliqués et plus ou moins commodes selon les machines.

Compte tenu . de la banalisation de la mémoire,
. de la compatibilité souhaitée avec les systèmes

sans partage de temps et sans l'option de protection-translation,

. de la simplicité recherchée pour le matériel,

la mémoire a été divisée artificiellement, vue du côté de l'unité centrale, en blocs de 1024 mots (2^{10}). Le numéro du premier bloc adressable, convenablement cadré, sert d'adresse de base de l'utilisateur, toutes les références à la mémoire, d'un programme en mode utilisateur, sont faites à partir de cette adresse de base, l'unité centrale travaillant, elle, sur des quantités relatives à (et indépendantes de) cette adresse de base. L'organe optionnel qu'est ce registre de base est inaccessible à l'utilisateur en mode utilisateur car c'est au moyen d'une instruction privilégiée (d'entrée-sortie) que ce registre est chargé par le superviseur. Les possibilités d'adressage sont bornées inférieurement (adresse 0) par la base et supérieurement par la vérification que le déplacement ne dépasse pas le contenu d'un second registre, dit de protection, chargé du nombre de blocs (de 1 K) autorisés (moins un pour comparer au déplacement tronqué). En fait, les adresses (déplacements) Ø à 17 ne sont pas translatées car elles indiquent les registres généraux et index, en général d'une technologie plus rapide, situés aux adresses absolues Ø à 17, dont tous les utilisateurs doivent pouvoir bénéficier. L'avantage de ce système est sa simplicité, avec 2 registres de 8 bits, un additionneur (8 bits + concaténation du déplacement modulo 2^{10}) et un comparateur (8 bits), il est possible d'isoler n'importe quel segment d'un nombre entier de K situé à partir d'une adresse multiple entière de K, pour le mode utilisateur, dans la mémoire toujours accessible sans restriction en mode superviseur comme dans les installations sans protection ni translation.

Avec ce système, les programmes sont réimplantables par simple déplacement d'un nombre entier de K, sans modification du contenu (les adresses n'étant que des déplacements par rapport à la tête du segment) même en cours d'exécution de programme par interruption prioritaire par exemple.

La gestion n'est sans doute pas aussi fine qu'avec un système à pages ou à mémoires associatives car les segments sont d'un seul tenant en mémoire centrale mais peut n'être pas trop coûteuse si des instructions rapides permettent des mouvements de zones de mémoire de façon à éliminer les trous que constituent les K inoccupés quand plusieurs programmes sont implantés dans la mémoire centrale et disparaissent indépendamment.

2 - Les seconds registres et la réentrance

La simplicité de ce mécanisme a encouragé sur certaines machines une extension avec l'option d'un second couple de registres de translation-protection pour pouvoir à un instant donné adresser 2 segments disjoints (ou non).

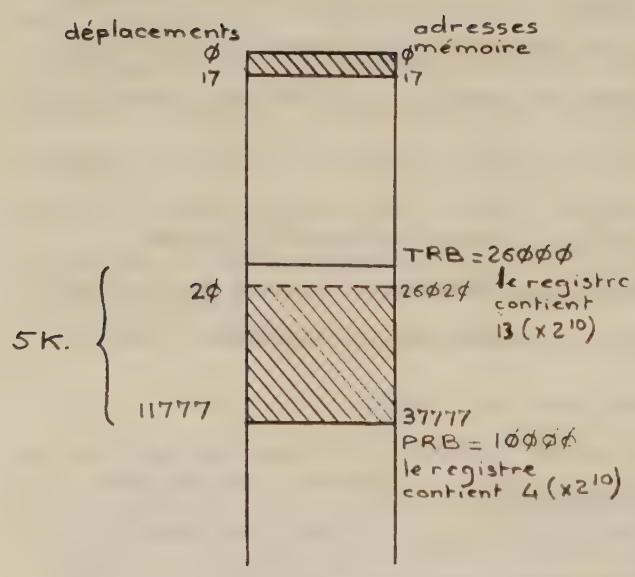
La programmation se trouve bien entendu compliquée ainsi que la gestion par le superviseur de partage de temps, chargé de veiller à davantage de conditions. Cet inconvénient est largement compensé dans le cas où ce second segment adressable peut bénéficier de façon simple de la garantie d'inaltérabilité, l'utilisateur ne pouvant rien y écrire mais seulement y lire et y exécuter des instructions. Si ce segment est inaltéré, il peut resservir de façon sûre après usage et en partage de temps, être utilisé par tranches de temps par plusieurs utilisateurs sans que ceux-ci sachent que leur zone de code ne leur est pas propre mais est partagée.

Ceci permet dans un système en partage de temps d'éviter d'implanter pour plusieurs utilisateurs simultanés d'un même programme, de longues zones de code identiques, qui resteront certainement inaltérées.

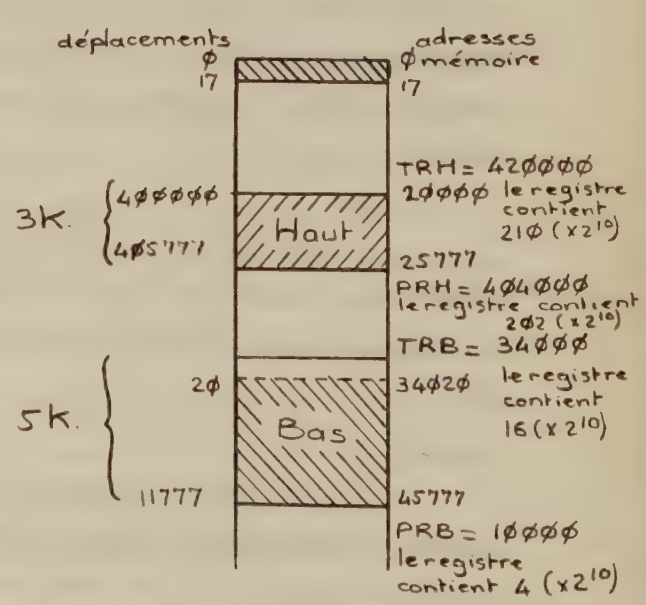
Pour pouvoir reconnaître lequel des 2 segments indique une adresse (déplacement) répartie entre Ø et $2^{18}-1$, il faut une convention simple et qui ne soit en rien restreignante pour les installations disposant d'une option à un seul segment et de programmes déjà écrits, de taille quelconque.

La convention adoptée ici est qu'un déplacement supérieur ou égal à 2^{17} (bit de plus fort poids du déplacement = 1) indique l'utilisation du second couple de registres sauf si ce déplacement est toléré par la protection du segment normal (premier registre de protection). Ainsi un segment du second type, nommé segment réentrant ou segment haut puisque les déplacements y faisant référence sont très élevés, ne saura dépasser en longueur la moitié de la capacité maximale en mémoire centrale (128 K), tandis que les segments normaux dits segments bas, peuvent toujours s'étendre de 1 à 256 K avec ou sans segment haut associé. Le segment haut est protégé par un registre de protection dont le contenu s'il est nul ou inférieur à 2^{17} ou à la protection basse, empêche toute référence avec une grande adresse à un segment haut.

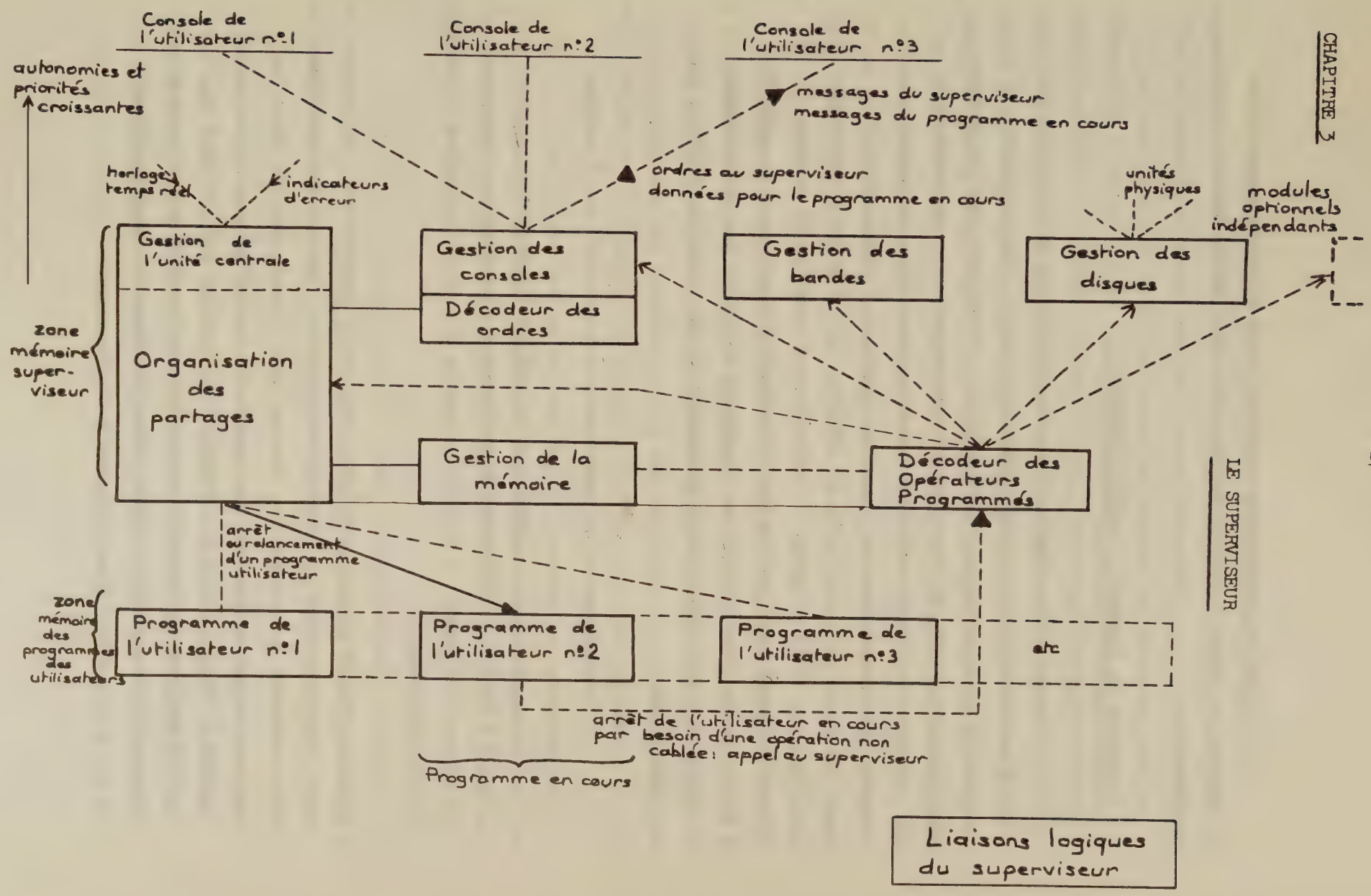
Lorsque s'applique le second type de translation, il est possible, s'il s'agit d'une opération d'écriture en mémoire et si un verrou a été positionné au chargement des 4 registres, de provoquer une interruption identique à une violation de protection.



Système à un segment
(non réentrant)



Système à 2 segments
(réentrant)



Liaisons logiques du superviseur

L'étude du superviseur va être faite de manière analytique en suivant la suite chronologique d'utilisation des différents modules constitutifs.

I - DEMARRAGE DU SYSTEME - INITIALISATIONS

Partant d'une mémoire d'un seul tenant du point de vue de l'unité centrale, cette dernière étant en état de fonctionnement, l'action sur la touche d'autochargement de la machine entraîne la lecture sur l'équipement sélectionné par des interrupteurs correspondants à son adresse sur la voie omnibus d'entrée-sortie, d'un programme autochargeur (en haut de mémoire) destiné à refaire l'implantation du superviseur mis en garde après sa création sur un support extérieur sous forme de fichier normal. L'autochargeur contrôle la lecture du fichier "superviseur", en réalise l'implantation et passe le contrôle à l'adresse de départ. C'est un programme exécutif, non translatable et spécialisé pour une sorte d'équipement et un format de fichier sur cet équipement.

L'initialisation du superviseur est exécutée par 2 modules

- l'un permanent, qui sera réutilisé en cas de réinitialisation du système à la suite d'un ennui grave qui n'a toutefois pas nécessité un nouveau chargement ; il fait la remise à jour de certaines tables et des équipements.

- l'autre utilisé une seule fois est destiné à acquérir des renseignements indispensables au lancement du système : date et heure doivent être demandées à l'opérateur, lues et leur format vérifié. Dans les systèmes munis de disques, les données essentielles quant à la structure et à l'emplacement des fichiers répertoires doivent être lues sur les disques eux-mêmes et en cas d'impossibilité de disposer de ces éléments sous un format correct, une procédure de réinitialisation de ces disques doit être proposée à l'opérateur.

Ce second module est placé dans les dernières adresses du superviseur ; après usage, les instructions qui l'ont appelé sont modifiées en non-opérations et son emplacement récupéré pour l'usage général. Il a également pour fonction de lancer le chainage de certains modules (mécanisme de recherche de cause d'interruption prioritaire) et de certaines tables enchaînées (état de chaque équipement) situées dans des modules différents ou même de créer de toutes pièces ces tables dans des zones libres à partir de renseignements fournis par le premier module issu du fichier définissant la CONFIGURATION (voir en annexe : Construction d'un superviseur). Ce travail est plus facile à faire par programme que par références globales et pointeurs dans les modules eux-mêmes, censés être indépendants entre eux, au moment du chargement et édition des liens.

Chaque équipement périphérique est ensuite initialisé grâce à la chaîne des tables d'état d'équipement, chaque table contenant entre autres, l'adresse dans le module de gestion de cet équipement, d'une table d'aiguillage par fonction. Parmi ces fonctions se trouve l'initialisation de l'équipement et de ses registres physiques propres.

L'initialisation se termine par le lancement du programme de l'utilisateur Ø, programme d'oisiveté incrémentant indéfiniment le registre Ø en attendant qu'un autre programme-utilisateur soit exécutable.

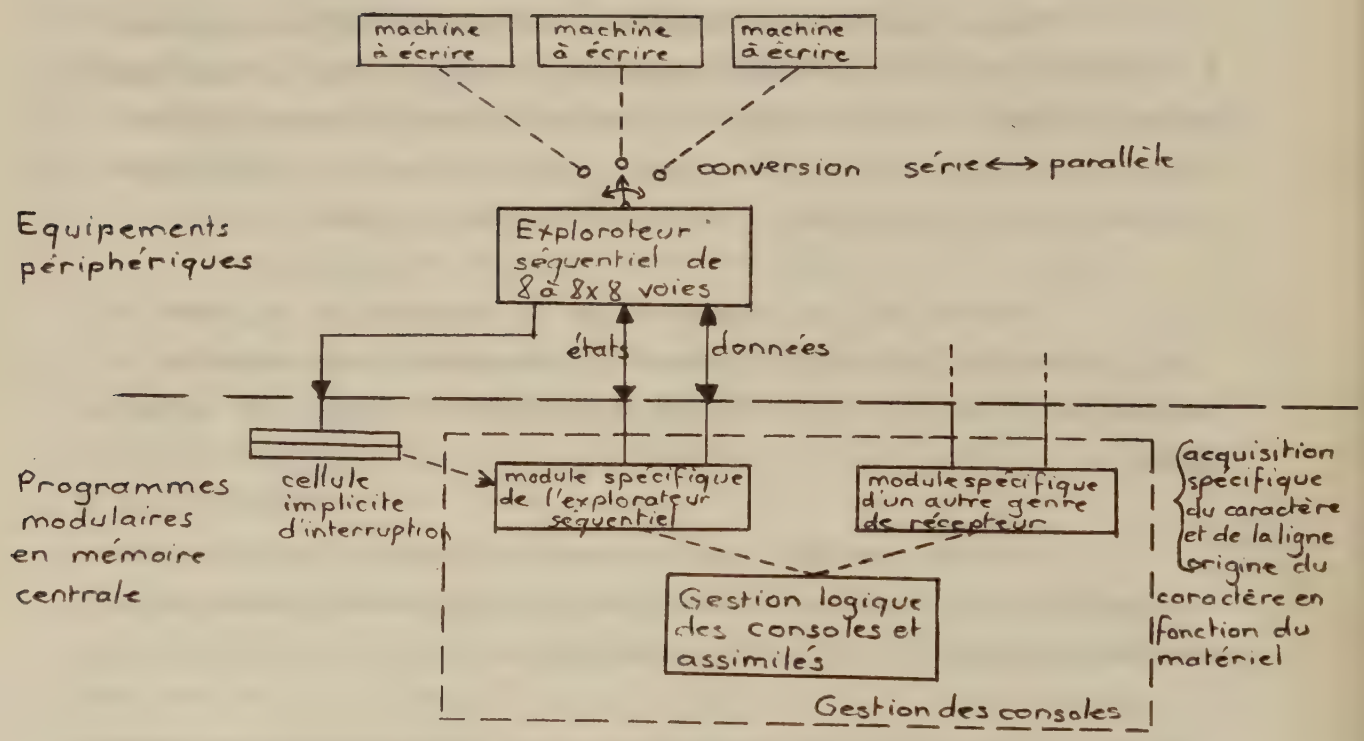
La tâche d'attente lancée en mode utilisateur sans allocation de mémoire si ce n'est les 16 registres généraux ne peut cesser que par interruption prioritaire. Celle-ci peut être :

- l'interruption régulière par l'horloge de l'unité centrale, interruption qui a pour effet global une analyse de la situation des différentes tâches possibles avec éventuellement interruption (non reprise) de l'utilisateur courant au profit d'un autre programme-utilisateur plus prioritaire,

- l'interruption aléatoire par frappe d'un ordre au superviseur sur une console d'utilisateur, les caractères accumulés forment un ordre quand le caractère de fin de message, ici de fin de ligne, est reçu.

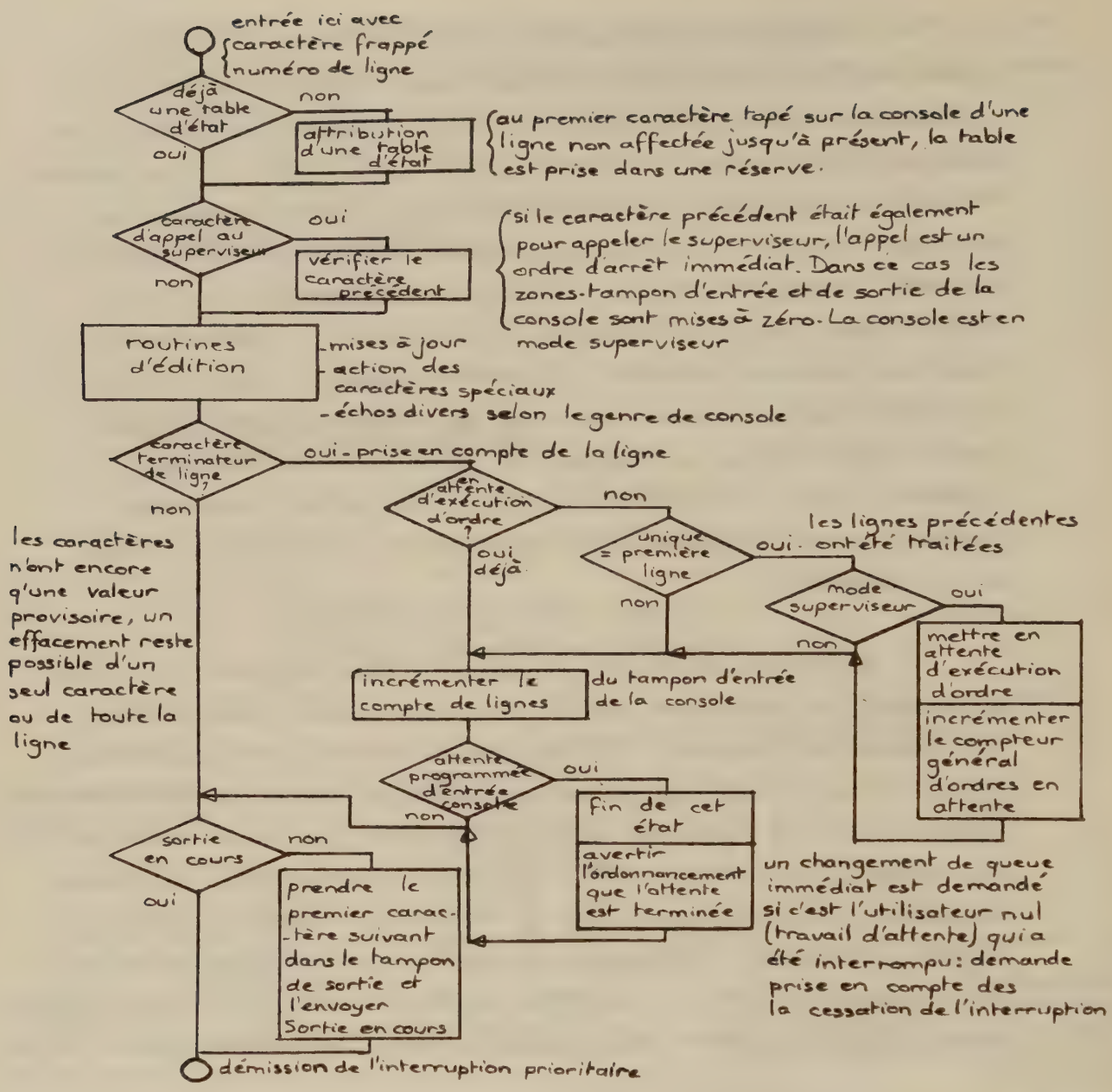
II - LA GESTION DES CONSOLES

II - LA GESTION DES CONSOLES



Tout ce qui concerne une console (ou machine à écrire) ou organe réel ou virtuel assimilé à une console susceptible d'envoyer au superviseur des ordres rédigés, c'est-à-dire tout utilisateur, est mémorisé dans une zone spécifique appelée table d'état de l'équipement. Cette table appartient à la chaîne des tables d'état, chaque console en possède une après la frappe du premier caractère, chaque dérouleur de bande, chaque périphérique particulier doit posséder une telle table d'état. Lors de la réception d'un caractère, l'accès à cette table est faite par recherche d'un numéro logique de ligne à partir du numéro physique indexant une table de caractéristiques physiques, lequel numéro logique en indexant une table de caractéristiques logiques fournit l'adresse de la table d'état et des indicateurs logiques comme l'attente d'exécution d'ordre ou le numéro de tâche de l'utilisateur.

Les tables d'état des consoles ne sont attribuées qu'en cas de demande, la pénurie est un cas prévu pour refuser temporairement de nouveaux utilisateurs, plus de lignes étant réellement existantes que de travaux simultanés envisagés à la définition de la CONFIGURATION (économie sur les tables et zones de rangement, crainte de surcharge du système). La table d'état est retournée au bien commun à l'abandon du système par un utilisateur ou par un ordre de désaffectation de la console, donné par l'utilisateur au superviseur. Cet ordre permet de poursuivre le travail sur une autre console au moyen d'un ordre de réaffectation et d'arguments judicieux pour identifier le demandeur.



Gestion des consoles

Comme il est indiqué sur l'organigramme et sachant qu'à l'état initial la console est considérée comme étant en mode superviseur, la première ligne frappée provoquera l'incrémentation du compteur général d'ordres en attente, ce compteur sera testé à la prochaine impulsion de l'horloge temps réel.

III - L'INTERRUPTION D'HORLOGE ET LES TACHES DIFFEREES

1 - Le report en priorité basse

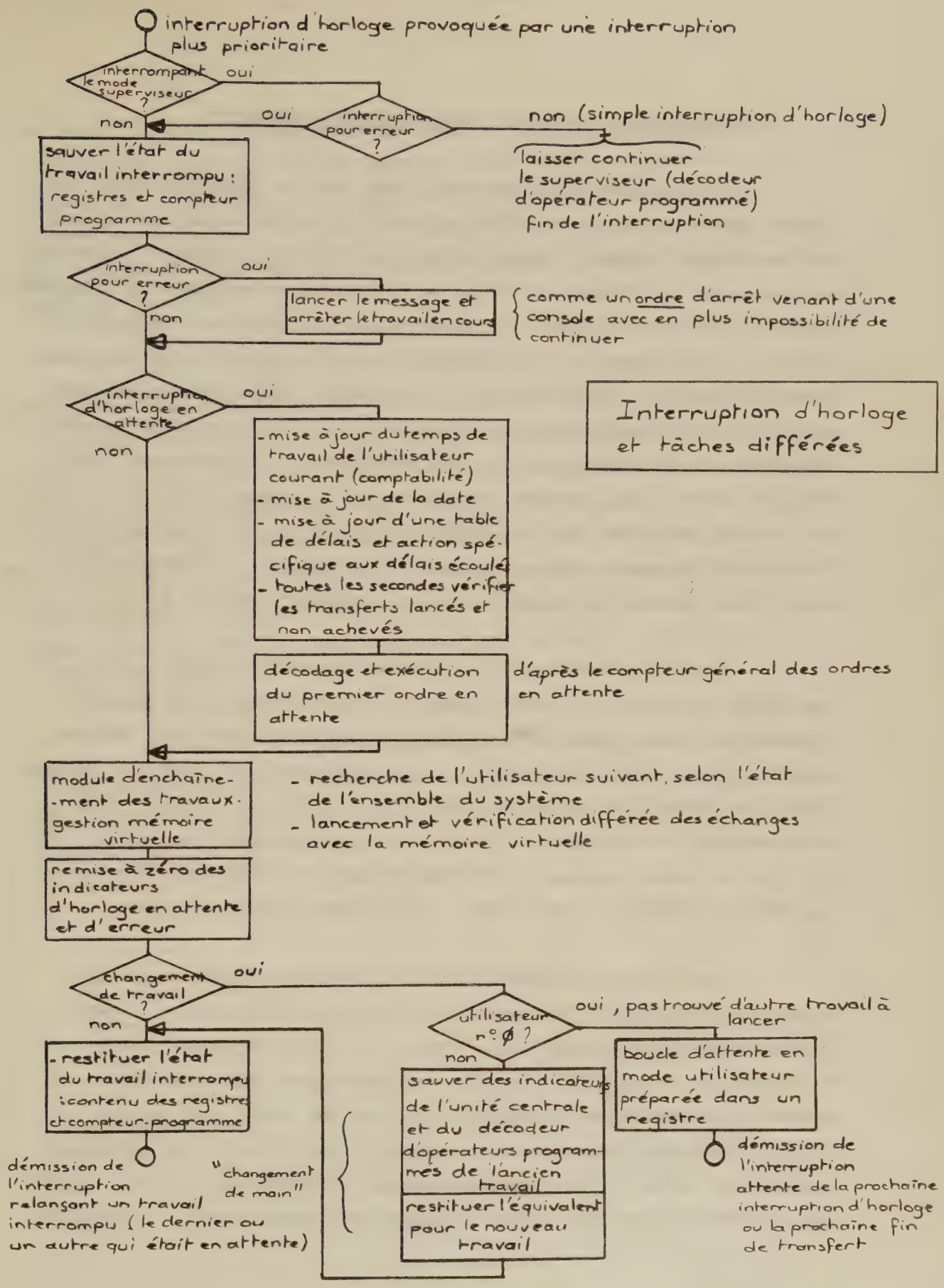
L'impulsion de l'horloge temps réel provoque une demande d'interruption très prioritaire acquittée en tant que telle par la routine d'identification des interruptions causées par l'unité centrale.

L'usage de cette impulsion hormis la tenue par simple incrémentation de l'heure du superviseur, n'est pas très prioritaire, au contraire, son but est de déclencher régulièrement l'exécution des différentes tâches différées du superviseur.

C'est pour cette raison que c'est au niveau le moins prioritaire qu'une interruption est demandée par programme à partir du niveau très prioritaire d'interruption de l'unité centrale ou plus exactement de ses indicateurs d'état anormal ce qui n'interrompt en rien son fonctionnement. Cette interruption doit cesser d'autant plus vite qu'elle est plus prioritaire et inhibe des possibilités d'échanges sur interruption moins prioritaire.

Le travail différé qui doit être exécuté régulièrement sur impulsion d'horloge est toujours moins prioritaire que les opérations d'entrée-sortie. Ce travail a pour but l'enchaînement des travaux en cours et l'exécution de tâches risquant d'être longues et ne devant pas entraver les mécanismes d'échanges avec les équipements d'entrée-sortie ; ceci explique le plus faible niveau des travaux de l'horloge, dans la hiérarchie des priorités d'interruption.

A défaut d'indicateur physique provocateur de l'interruption et reconnaissable par les instructions de test d'état de l'équipement physique, un drapeau programmé est positionné au niveau très prioritaire pour être testé à la demande d'interruption au niveau dit "de l'horloge", le moins prioritaire.


interruption d'horloge provoquée par une interruption plus prioritaire
interrompant le mode superviseur ?
oui
non
oui
interruption pour erreur ?
non (simple interruption d'horloge)
laisser continuer le superviseur (décodeur d'opérateur programmé) fin de l'interruption
sauver l'état du travail interrompu : registres et compteur programme
interruption pour erreur ?
oui
non
lancer le message et arrêter le travail en cours
comme un ordre d'arrêt venant d'une console avec en plus impossibilité de continuer
interruption d'horloge en attente
oui
non
Interruption d'horloge et tâches différées
- mise à jour du temps de travail de l'utilisateur courant (comptabilité)
- mise à jour de la date
- mise à jour d'une table de délais et action spécifique aux délais écoulés
- toutes les secondes vérifier les transferts lancés et non achevés
décodage et exécution du premier ordre en attente
d'après le compteur général des ordres en attente
module d'enchaînement des travaux - gestion mémoire virtuelle
- recherche de l'utilisateur suivant, selon l'état de l'ensemble du système
- lancement et vérification différée des échanges avec la mémoire virtuelle
remise à zéro des indicateurs d'horloge en attente et d'erreur
changement de travail ?
oui
non
utilisateur n° Ø ?
oui, pas trouvé d'autre travail à lancer
non
- restituer l'état du travail interrompu : contenu des registres et compteur-programme
sauver des indicateurs de l'unité centrale et du décodeur d'opérateurs programmés de l'ancien travail
restituer l'équivalent pour le nouveau travail
boucle d'attente en mode utilisateur préparée dans un registre
démission de l'interruption relançant un travail interrompu (le dernier ou un autre qui était en attente)
"changement de main"
démission de l'interruption attente de la prochaine interruption d'horloge ou la prochaine fin de transfert

2 - L'interblocage avec les tâches en mode exécutif

Une considération importante est que des tâches différées ne doivent pas être exécutées prioritairement à des tâches du superviseur déjà en cours, lorsque l'interruption d'horloge est survenue. La raison est que parmi ces tâches différées se trouvent le changement de priorités relatives des utilisateurs du système selon un algorithme équitable et surtout le changement de l'utilisateur courant consécutivement au changement de priorité. Le changement d'utilisateur est une chose relativement simple, les sauvetages sont bien définis. Il n'en est pas de même lorsque c'est le superviseur qui est interrompu, le superviseur peut contenir ça et là des données nécessaires à la poursuite de son travail : par exemple lors d'un appel au superviseur, par un programme utilisateur, l'adresse 40 absolue contient le code instruction de l'opérateur programmé mais comment le savoir ? vu d'un niveau prioritaire ?

Face à ce problème, il a été décidé de différer un peu plus les tâches en attente, de faire cesser l'interruption et de laisser continuer le superviseur dans son travail, à charge pour lui, dès que possible, c'est-à-dire dès qu'il ne peut plus continuer faute de données (après lancement d'un ordre de transfert) ou qu'il doit relancer le programme utilisateur qui fit appel à lui par opérateur programmé, s'il voit que le niveau horloge a été demandé mais non acquitté, d'effectuer les tâches différées à son niveau, c'est-à-dire non prioritaire.

Les tâches différées sont donc exécutées tantôt au niveau un peu prioritaire de l'horloge (si au moment de l'impulsion, c'est d'un programme utilisateur que s'occupait l'unité centrale) tantôt au niveau non prioritaire, normal, du superviseur après une petite attente supplémentaire. Eventuellement pour une tâche très longue une interruption d'horloge arrivera avant d'avoir traité la précédente (perte d'un cycle), c'est sans importance, l'indicateur de temps ayant, lui, été mis à jour sans tarder à chaque interruption très prioritaire.

La raison de cet interblocage entre les 2 niveaux du superviseur est l'imparfaite réentrance du décodeur d'opérateurs programmés et des modules qu'il appelle. Il y a également quelques ordres dont l'exécution, très longue, implique des transferts de données et, comme telle, est analogue au décodage des opérateurs programmés (voir appel d'un programme externe).

Cette non-interruption du superviseur en travail pour un utilisateur dont il décode un opérateur programmé, pour satisfaire les tâches différées sous interruption d'horloge disparaît dans le cas où c'est une erreur détectée par l'unité centrale qui a causé l'interruption première mais dont l'analyse des causes et l'avertissement de l'utilisateur concerné ont été différés au niveau horloge. De toute façon, l'utilisateur courant pour lequel le superviseur a fait une erreur grave et imprévue ne continuera pas la tâche en cours : avertissement et arrêt en attendant un nouvel ordre (mise en queue d'arrêt).

Au départ, au premier lancement de travail, c'est le module chargé du décodage et de l'exécution du premier ordre en attente qui est déterminant. On n'y fait appel que si le compteur d'ordres en attente n'est pas nul.

3 - Le décodage et l'exécution des ordres

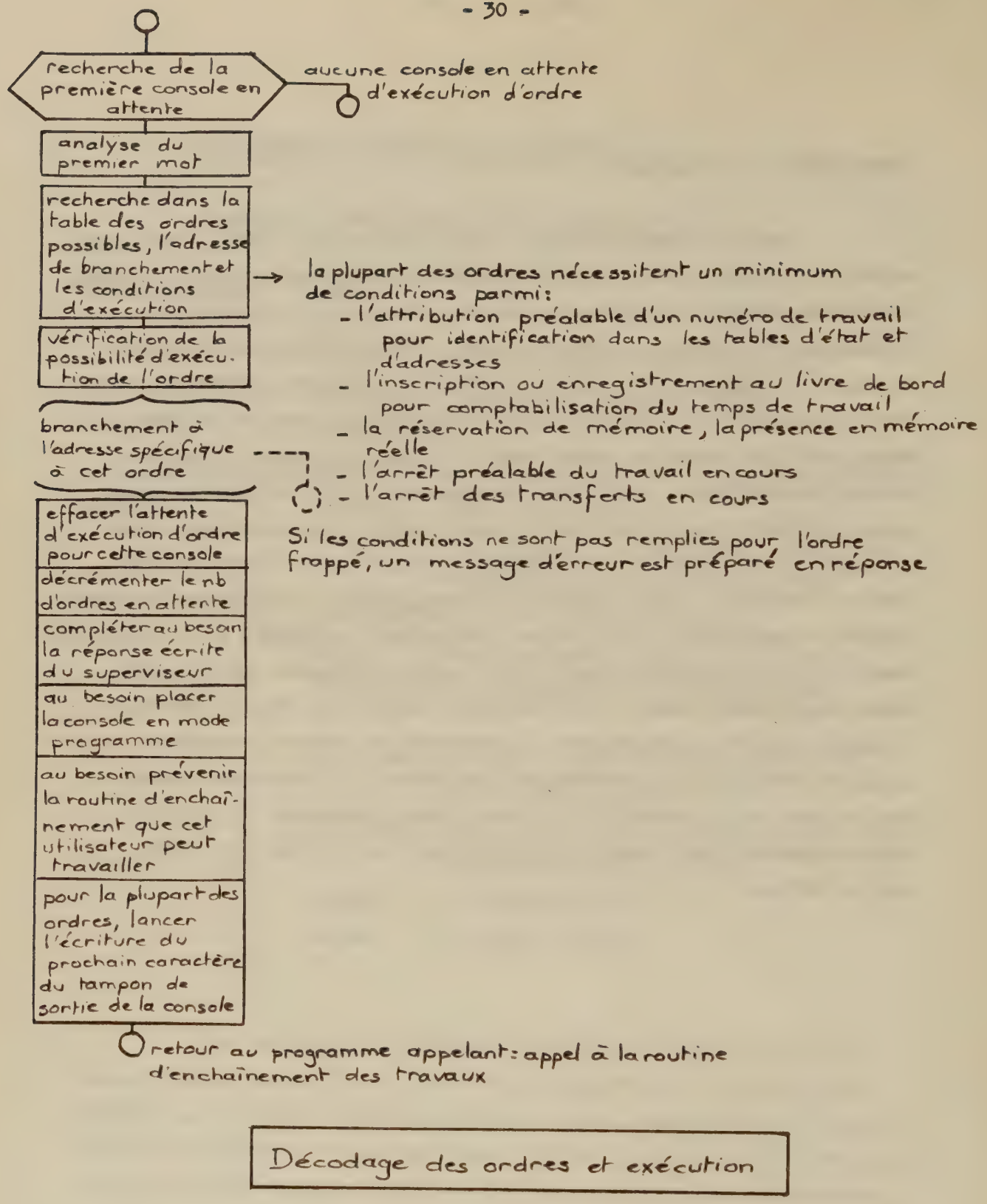
Exemples d'ordres destinés au superviseur :

renseignements :	HEURE
	DATE
	TEMPS (écoulé en occupant l'unité centrale)
	RESSOURCES (partageables encore libres)
	ETAT du système
réservations :(m')	ATTRIBUER tel périphérique (avec tel nom symbolique)
exécution :	MEMOIRE nombre de K désirés
	RECHERCHE (et expansion en mémoire d'un programme dont on donne la source et le nom)
	LANCEMENT avec une adresse éventuelle
	ARRET (forcé sur ordre avant fin naturelle)
	SAUVEGARDE(du programme sur tel périphérique)
	des ordres combinés : pour les programmes de la bibliothèque système
	des ordres pour exécuter des programmes implicites de la bibliothèque système
	ARRIVEE (m'inscrire à bord)
	DEPART (me rayer du bord)

Les consoles des utilisateurs servent d'abord de moyen de communication entre l'utilisateur et le superviseur. Ce dernier est capable de s'effacer (sauf cas d'erreur) et de permettre le dialogue entre l'utilisateur et son programme, lequel programme peut à son tour établir une distinction entre ses ordres propres et les informations qu'il traite sans entrer dans leur détail : ainsi l'utilisateur dit au créateur général de fichier quel fichier il va créer, puis commence à écrire son fichier à son gré, il avertira qu'il a fini et qu'il faut que le programme créateur ferme ce fichier par un code spécial conven-

tionnel dans ce programme.

Pour le superviseur les choses sont moins nuancées, une ligne entrée à la console est, ou bien destinée à être décodée comme un ordre, ou bien destinée au programme en cours et donc ne le concerne pas. L'importance de chaque caractère frappé, l'acquisition par caractère, le besoin de regroupement par mot machine et la double nature des informations reçues font qu'un tampon est indispensable dans le système, au moins pour les ordres au superviseur. Ceci est à opposer à tous les autres transferts qui sont faits directement du périphérique vers des zones baptisées tampons et dans le sens inverse. Ces zones sont dans le programme de l'utilisateur concerné quelque soit le moyen utilisé pour ce transfert (unité centrale ou canal direct), de plus ce transfert peut être asynchrone, c'est-à-dire concomitant du travail de l'utilisateur. Ceci est encore compliqué par le fait qu'un utilisateur peut vouloir taper des lignes d'avance dont certaines seront des ordres, car elles n'auront pas été demandées par le programme en cours au moment où celui-ci fera appel au superviseur pour dire qu'il a terminé auquel cas le mode de la console n'est plus le mode programme. Ce genre de synchronisation des lignes n'est d'ailleurs par parfois sans être mis en défaut.



Décodage des ordres et exécution

4 - Exemple d'ordre : charger un programme et le lancer

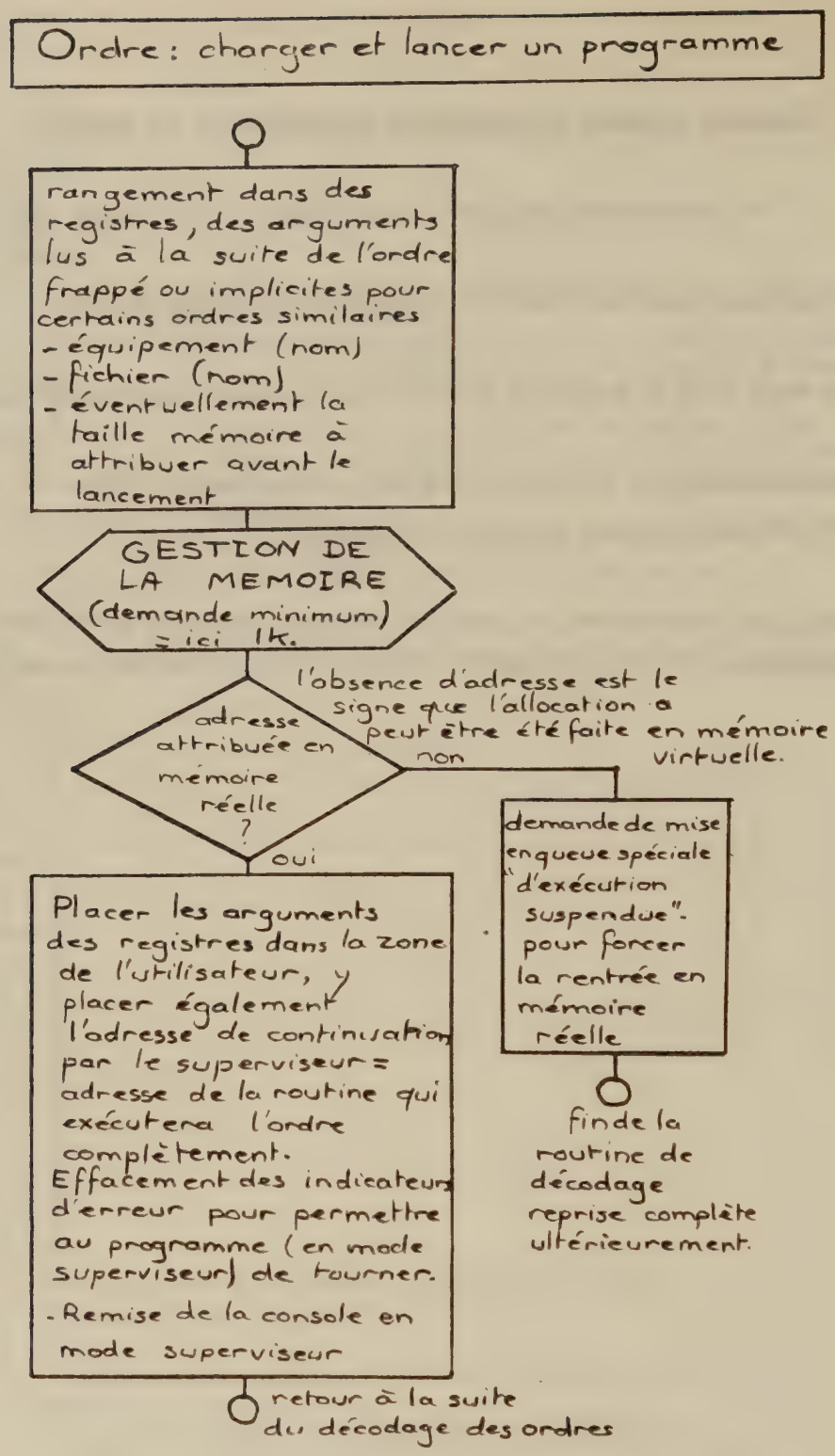
Les conditions requises pour exécuter cet ordre sont :

- le programme précédent de l'utilisateur doit être arrêté,

- il ne doit plus y avoir de transferts en cours (par routines prioritaires),

- la zone mémoire de l'utilisateur doit être immobile (pas en cours de recopie en mémoire réelle ou virtuelle),

moyennant ces conditions et sans qu'une zone mémoire préalable soit indispensable, la routine spécifique d'exécution de cet ordre est :


Ordre : charger et lancer un programme
rangement dans des registres, des arguments lus à la suite de l'ordre frappé ou implicites pour certains ordres similaires
- équipement (nom)
- fichier (nom)
- éventuellement la taille mémoire à attribuer avant le lancement
GESTION DE LA MEMOIRE (demande minimum) = ici 1k.
adresse attribuée en mémoire réelle ?
l'absence d'adresse est le signe que l'allocation a peut être été faite en mémoire virtuelle.
non
oui
demande de mise en queue spéciale "d'exécution suspendue". pour forcer la rentrée en mémoire réelle
fin de la routine de décodage reprise complète ultérieurement.
Placer les arguments des registres dans la zone de l'utilisateur, y placer également l'adresse de continuation par le superviseur = adresse de la routine qui exécutera l'ordre complètement.
Effacement des indicateurs d'erreur pour permettre au programme (en mode superviseur) de tourner.
- Remise de la console en mode superviseur
retour à la suite du décodage des ordres

Cet ordre relativement long n'est pas exécuté en tâche différée, il est transformé en commande exécutable au compte de l'utilisateur, certes par le superviseur mais avec une priorité égale à celle des utilisateurs courants. Ce régime non prioritaire est d'ailleurs indispensable pour pouvoir utiliser les opérateurs programmés habituels et leur décodeur (non réentrant).

La recherche d'un programme faite par le superviseur doit être comprise comme une aide qu'il apporte à l'utilisateur désireux de recharger un programme exécutable rangé sur un support extérieur. Cette aide n'est d'ailleurs pas indispensable, des ordres plus simples existent qui sont :

- la réservation de mémoire,

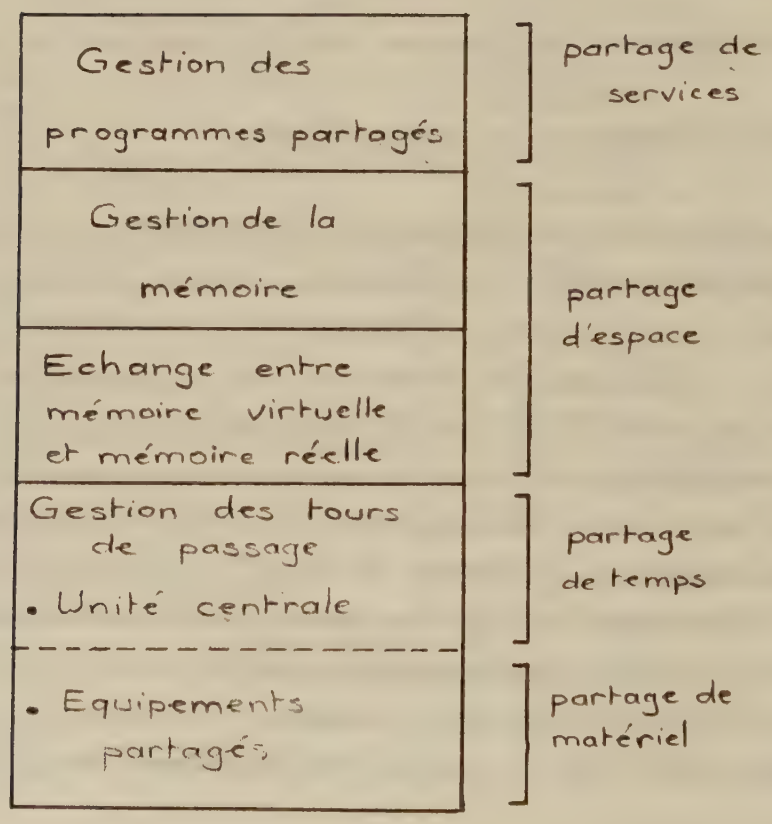
- le positionnement à une adresse octale, d'une quantité octale,

- l'ordre de démarrer à une adresse donnée,

rien n'empêcherait un utilisateur de positionner ainsi un petit programme chargeur appelant son programme principal au moyen d'opérateurs programmés d'entrée-sortie. La situation est tout à fait identique à celle de l'opérateur d'un ordinateur dont la mémoire (celle de l'ordinateur) est vide, il utilise un chargement télescopique à partir de quelques instructions déposées aux clés de la console ou bien dispose d'une touche d'autochargement à partir d'un équipement sélectionné mécaniquement. En partage de temps, cette aide que constitue l'ordre de chargement ne doit pas pour autant entraîner une priorité sur les utilisateurs interrompus pour traiter cette tâche différée. Le détail de la recherche et du lancement d'un programme par le superviseur figure dans la gestion des programmes réentrants.

IV - LES PARTAGES

Le premier exemple de tâche différée a nécessité l'appel à un programme capable d'attribuer à l'utilisateur concerné, une zone minimale de 1 K mots compte tenu de la situation globale de la mémoire du système. Pour ce faire, il y a donc eu partage d'une des ressources : la place en mémoire et c'est un des modules de l'organisation des partages qui a tenté cette attribution. Cette organisation des partages met en oeuvre plusieurs modules :



L'étude des différents modules est entreprise ci-dessous dans l'ordre où ils seront appelés pour le premier utilisateur du premier programme susceptible de tourner : gestion de la mémoire, échange entre mémoire virtuelle et mémoire réelle, gestion des programmes partagés, tours de passage en unité centrale et sur les équipements partagés.

V - GESTION DE LA MEMOIRE

La ressource "mémoire" est divisée en 2 parties :

1 - La mémoire réelle

La mémoire réelle est constituée d'un certain nombre de K (1024 mots) consécutifs (pour l'adressage absolu sur les portes de la V.O. Mémoire venant de l'unité centrale) situés entre la plus haute adresse du superviseur arrondie au K immédiatement supérieur et la plus haute adresse possible reliée sans interruption à l'adresse Ø compte tenu du branchement des bancs de mémoire à l'unité centrale. Cette taille est déterminée par le superviseur lui-même lors de son initialisation. L'occupation de cette mémoire est représentée par une table d'occupation à raison d'un bit par K pour indiquer son utilisation (Ø = libre)

2 - La mémoire virtuelle

La mémoire virtuelle est située sur un disque rapide à têtes fixes ou un tambour magnétique servant bien souvent également de support des fichiers d'utilisateurs et des fichiers de la bibliothèque système. Toute une partie de ce support, représentant en volume la somme de tous les programmes susceptibles de coexister à un instant donné pour tourner à tour de rôle est réservée au rangement temporaire des segments de programmes en attente de travailler, comme extension

de la mémoire réelle. L'occupation de cette mémoire est représentée par une table d'occupation située dans le supervisuer, à raison d'un bit par K. Un segment de programme constitué obligatoirement de K contigus en mémoire réelle peut ici être rangé par fragments (nombre entier de K) non contigus pour un meilleur remplissage de cette mémoire virtuelle.

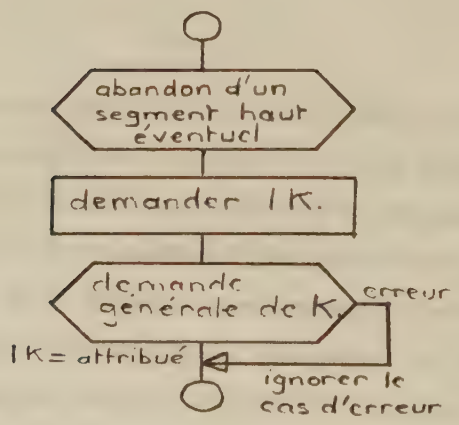
Les liens, entre les fragments, sont conservés temporairement en mémoire réelle (zone banalisée du superviseur) pendant le séjour du segment fragmenté en mémoire virtuelle.

(Le programme d'un utilisateur peut être décomposé en 1 ou 2 segments).

Initialement, aucun utilisateur ne dispose de mémoire, les tables d'occupation sont vides (mémoires virtuelle et réelle), les compteurs de K disponibles maximaux.

3 - Réservation de mémoire - Routine générale

En général, l'attribution de mémoire à un utilisateur est faite par un ordre d'appel de programme qui commence par une demande d'attribution minimale.

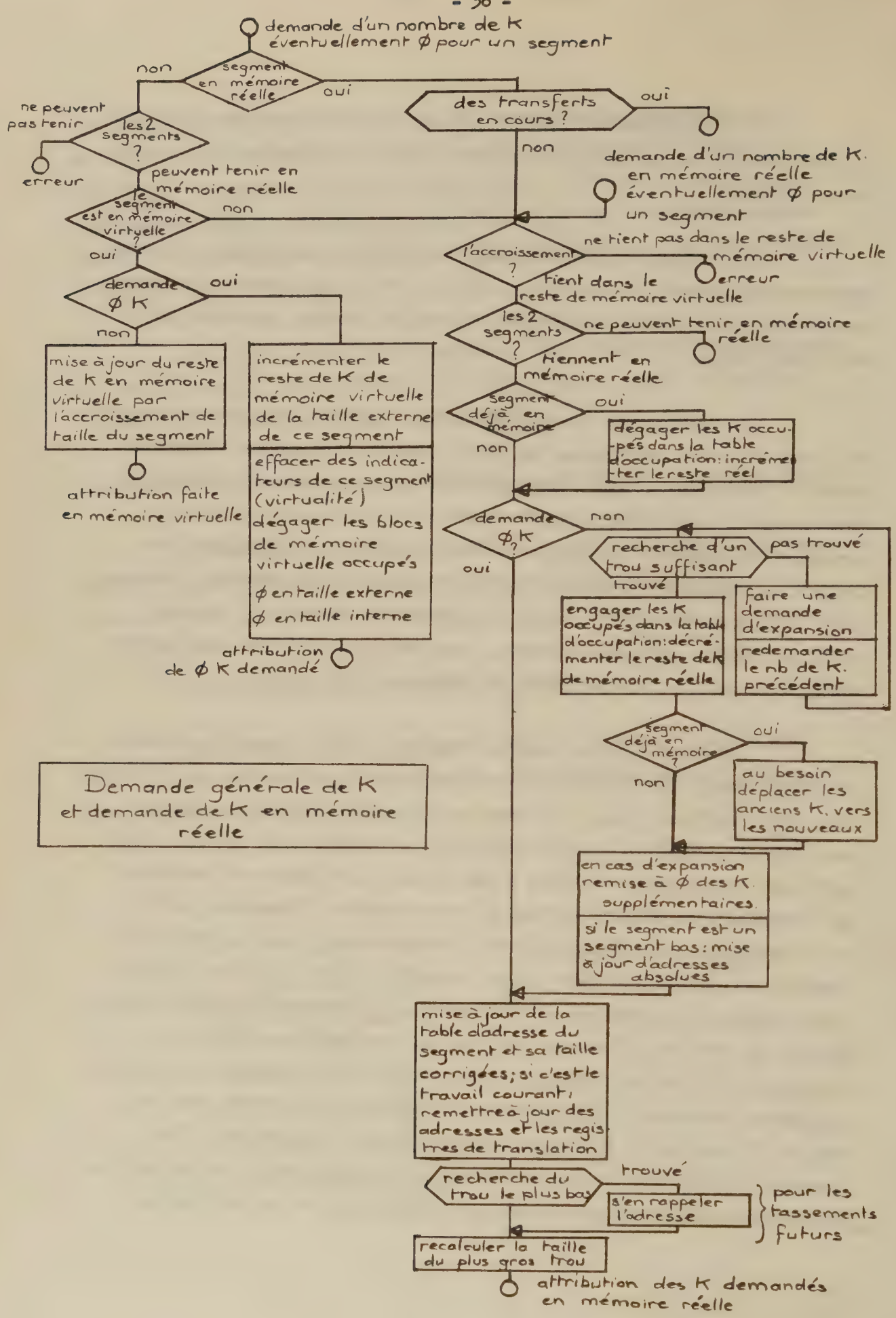


Si l'attribution minimale n'a pu être faite en mémoire réelle, l'ordre est mis en attente, c'est-à-dire que son exécution est de nouveau différée.

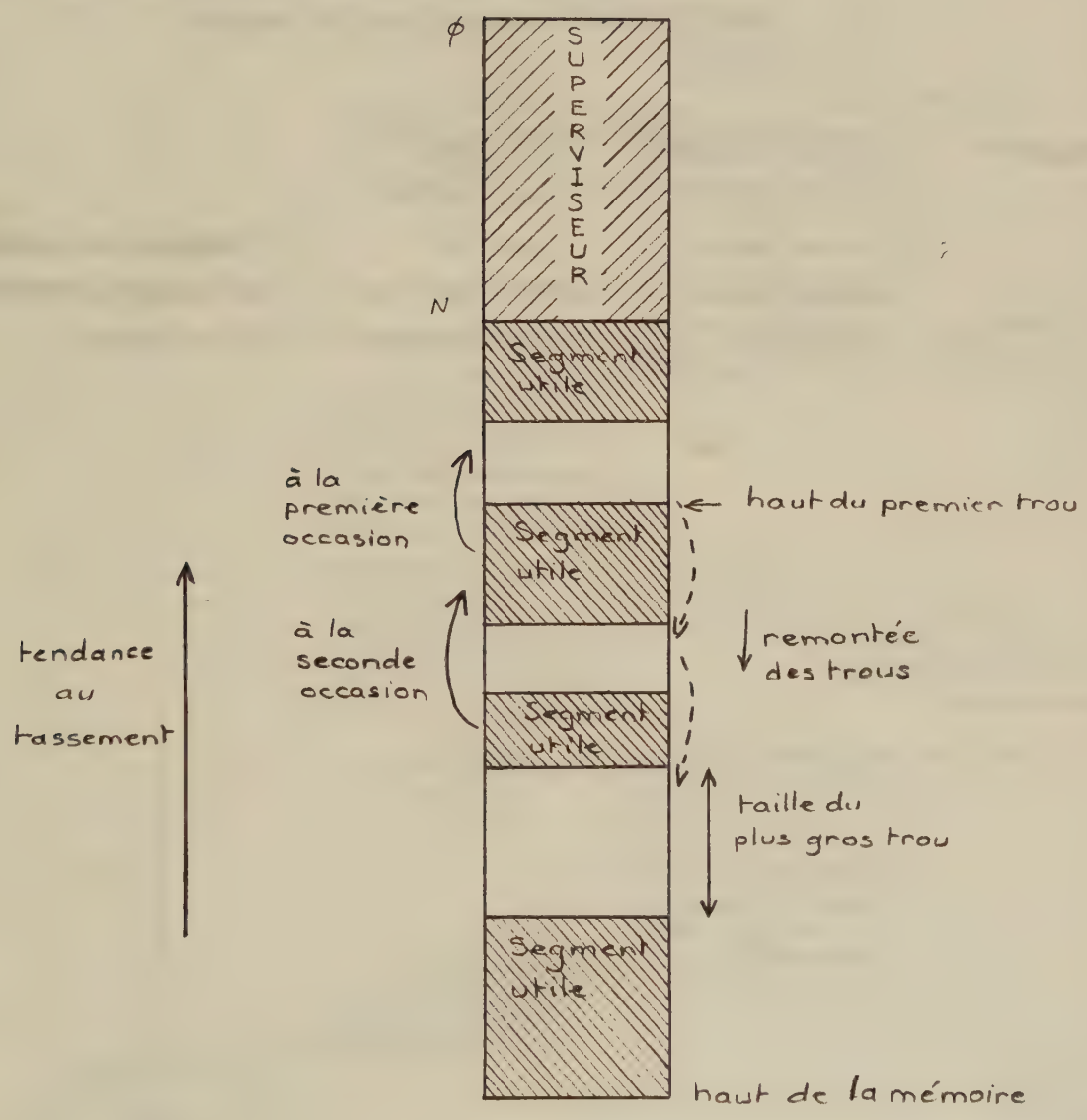
Le travail de l'utilisateur est mis dans une queue d'attente spéciale favorisant la rentrée en mémoire réelle du segment qui lui a sans doute été attribué en mémoire virtuelle (à moins que celle-ci ne soit déjà pleine).

La réservation de mémoire par la routine générale des demandes de K est faite de préférence en mémoire réelle, mais peut l'être en mémoire virtuelle, c'est l'absence d'adresse du segment au retour de cette routine qui signifie une attribution (fictive s'il s'agit de création de segment) en mémoire virtuelle qui ne se "réalisera" que lors d'un transfert en mémoire réelle. Lors de cette rentrée, l'expansion (ou la création) est faite simplement par recopie du segment dont on connaît la taille externe, dans une zone réservée plus grande égale à la taille interne (voir le mécanisme d'échange).

Lors des attributions de mémoire, outre l'abandon préalable de l'ancienne place éventuelle du segment dont on s'occupe, la recherche d'un trou suffisant pour le nombre de K demandés se fait en partant des adresses absolues les plus faibles, le premier trou (le plus bas) suffisant est occupé, si aucun n'est trouvé, une demande d'expansion par transfert en mémoire virtuelle est déposée et l'ancienne taille est redemandée, avec éventuellement un déplacement vers le bas de la mémoire, si un trou de même taille a pu être trouvé plus bas. Lors de chaque nouvelle allocation ou réallocation ou libération (réallocation de Ø K) de mémoire, l'adresse du trou le plus bas (ou plutôt de la fin de ce trou) et la taille du plus gros trou sont mémorisés pour faciliter le tassement des segments, pour déplacer les trous vers le haut de la mémoire et les segments vers le bas ; ainsi lors d'un transfert en mémoire réelle ou de façon générale lors d'une allocation, on a le maximum de chance de trouver un trou de taille suffisante.


demande d'un nombre de K éventuellement Ø pour un segment
segment en mémoire réelle
non
oui
des transferts en cours ?
oui
non
demande d'un nombre de K. en mémoire réelle éventuellement Ø pour un segment
les 2 segments ?
ne peuvent pas tenir
erreur
peuvent tenir en mémoire réelle
le segment est en mémoire virtuelle ?
non
oui
l'accroissement ?
ne tient pas dans le reste de mémoire virtuelle
erreur
tient dans le reste de mémoire virtuelle
demande Ø K
oui
non
les 2 segments ?
ne peuvent tenir en mémoire réelle
tiennent en mémoire réelle
mise à jour du reste de K en mémoire virtuelle par l'accroissement de taille du segment
incrémenter le reste de K de mémoire virtuelle de la taille externe de ce segment
effacer des indicateurs de ce segment (virtualité)
dégager les blocs de mémoire virtuelle occupés
Ø en taille externe
Ø en taille interne
attribution faite en mémoire virtuelle
attribution de Ø K demandé
segment déjà en mémoire
oui
non
dégager les K occupés dans la table d'occupation: incrémenter le reste réel
demande Ø K ?
non
oui
recherche d'un trou suffisant
pas trouvé
trouvé
engager les K occupés dans la table d'occupation: décrémenter le reste de K de mémoire réelle
faire une demande d'expansion
redemander le nb de K. précédent
segment déjà en mémoire ?
oui
non
au besoin déplacer les anciens K. vers les nouveaux
en cas d'expansion remise à Ø des K. supplémentaires.
si le segment est un segment bas: mise à jour d'adresses absolues
mise à jour de la table d'adresse du segment et sa taille corrigées; si c'est le travail courant: remettre à jour des adresses et les registres de translation
recherche du trou le plus bas
trouvé
s'en rappeler l'adresse
pour les tassements futurs
recalculer la taille du plus gros trou
attribution des K demandés en mémoire réelle
Demande générale de K et demande de K en mémoire réelle

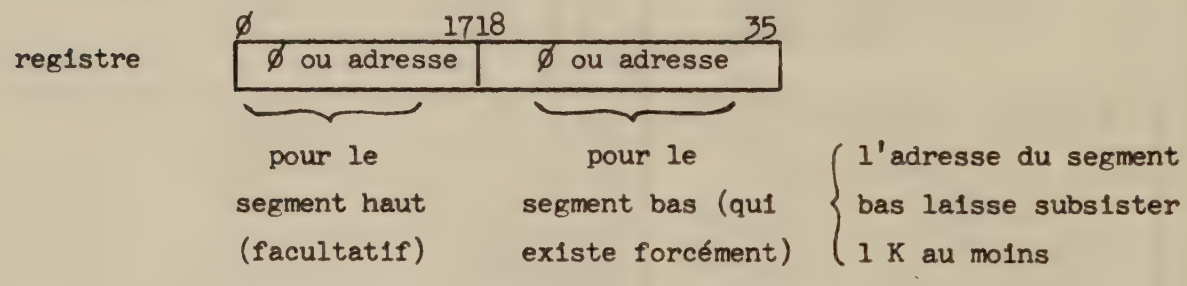
Les phénomènes de tassement dans une situation statique sont inutiles, ils ne seront mis en oeuvre que quand le plus gros trou ne suffira pas à satisfaire une demande d'allocation par la routine d'échange de mémoire virtuelle vers la mémoire réelle.

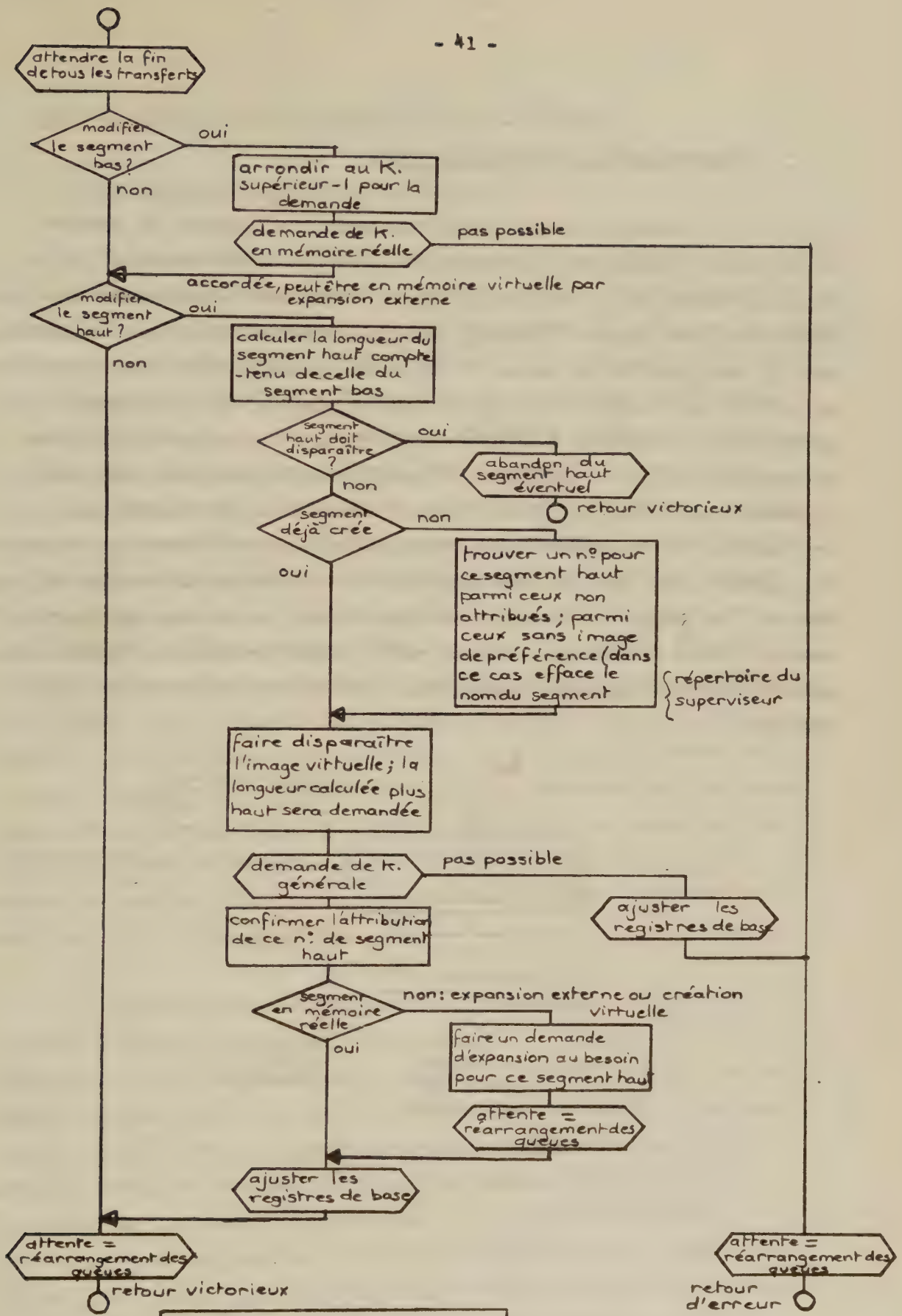


4 - Réservation de mémoire par opérateur programmé

L'allocation de mémoire est comme il a été vu jusqu'ici une conséquence du décodage dans certains ordres frappés sur une console. Cette allocation peut ensuite être modifiée par le programme lancé, lui-même, au moyen d'un opérateur programmé. Il suffit de passer en argument, dans un des 16 registres généraux deux moitiés de mots ayant pour signification Ø pas de changement au segment

≠ Ø plus haute adresse (à arrondir au K supérieur - 1) dont on veut disposer dans ce segment

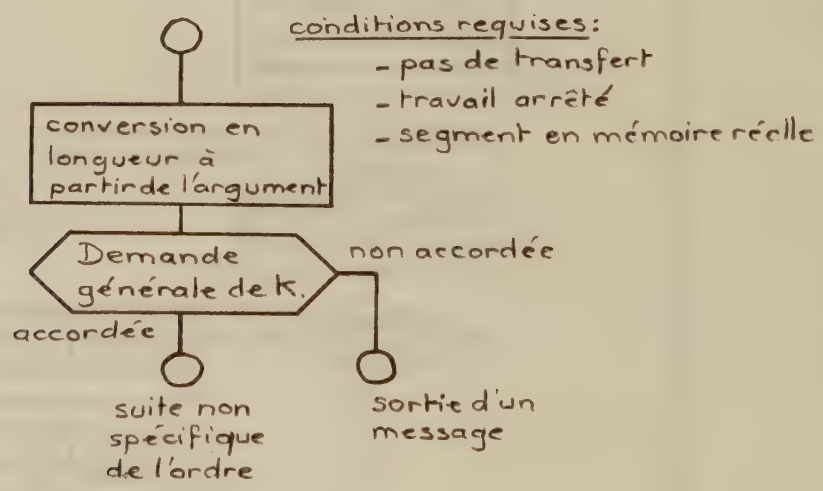




Allocation de mémoire par opérateur programmé

5 - Réservation de mémoire par ordre

Il existe un ordre spécialisé pour l'allocation de mémoire ; en l'absence d'argument, c'est une question qui a pour réponse l'allocation actuelle et le total de la place disponible ; en argument le nombre de K est passé en décimal, ceci ne concerne que le seul segment bas. Il est possible au moyen de cet ordre (le seul) d'annuler toute allocation de mémoire ; ce n'est en général pas le cas des travaux arrêtés, qui continuent à utiliser de la mémoire en attendant un ordre de continuation ou de lancement. L'appel d'un nouveau programme, par la réattribution minimale fait disparaître au moins le segment bas de l'ancien (et le segment haut au moins de la zone d'adressage de cet utilisateur) sans obliger une annulation de l'ancienne allocation de mémoire. Le langage de commande que constituent les ordres au superviseur doit être économique, un seul ordre pouvant remplacer d'autres ordres plus fins également autorisés, c'est le cas de la gestion de mémoire.

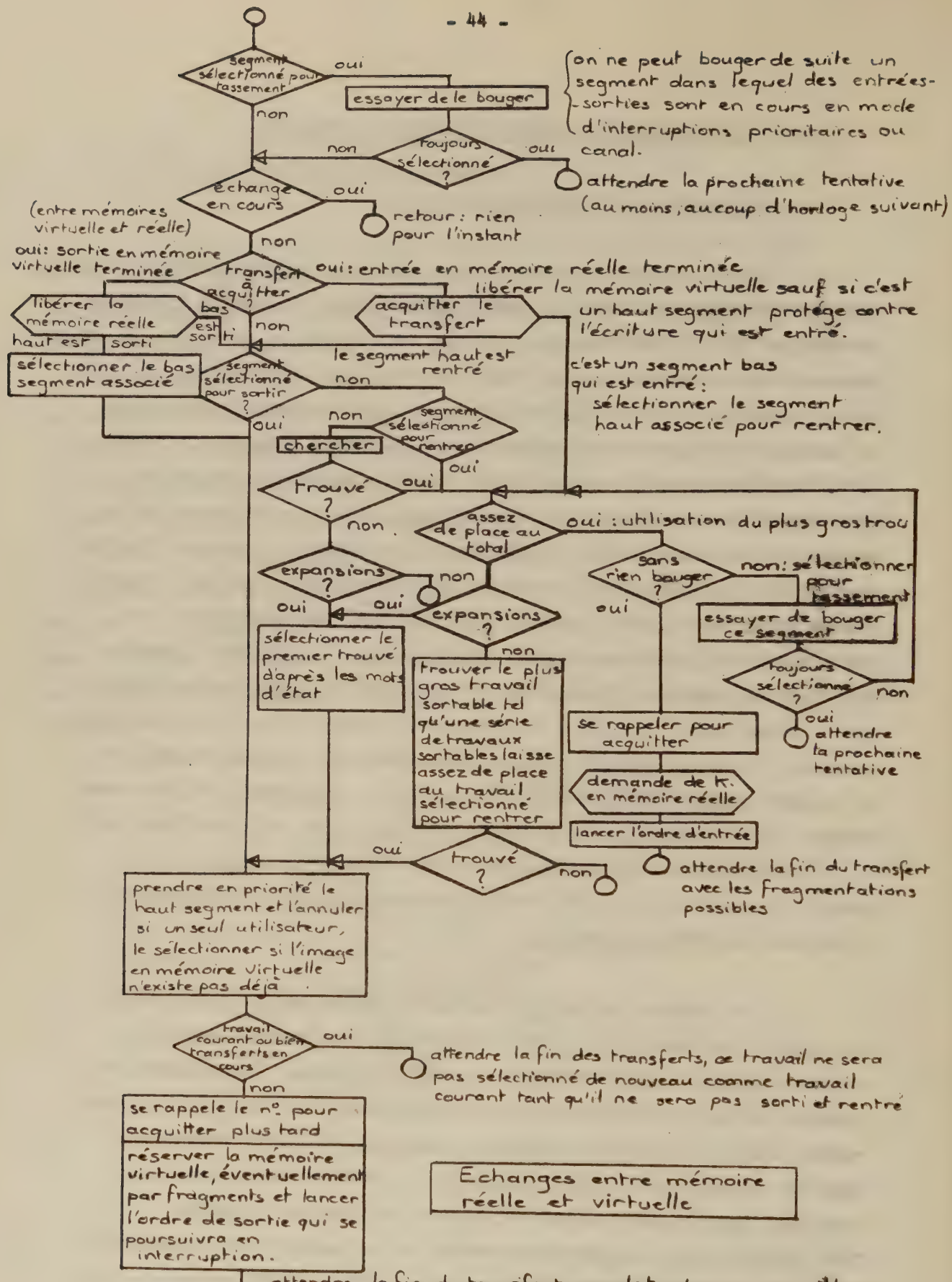


VI - ECHANGES ENTRE MEMOIRE REELLE ET MEMOIRE VIRTUELLE

1 - Organisation de la mémoire virtuelle

Une installation simple, sans équipements périphériques rapides, peut très bien fonctionner en temps partagé, avec un nombre limité de travaux, en n'utilisant pour ranger les programmes en cours que la mémoire rapide, d'accès sélectif mais très coûteuse, à sa disposition. La limitation essentielle provient de la taille des programmes individuels, le partage de la place est prépondérant et limite beaucoup le taux d'utilisation des autres ressources. Pour augmenter les capacités d'accueil d'un système en partage de temps, il est rapidement indispensable de pouvoir ranger les programmes en attente, sur une mémoire auxiliaire, de plus grande capacité que la mémoire centrale, d'accès rapide et autonome (pour que les transferts s'imbriquent avec les exécutions de l'unité centrale). Physiquement, la mémoire d'appoint baptisée ici, mémoire virtuelle, est un tambour ou un disque magnétique à têtes fixes, connecté par un canal d'accès direct à la mémoire centrale, elle est adressable par blocs. Dans le système étudié ici, le disque support de la mémoire virtuelle est également utilisé comme support d'une structure de fichiers et l'usage de son contrôleur est à ce titre, partagé entre la gestion des fichiers et celle de la mémoire virtuelle implantée sur des pistes réservées avec une priorité pour la mémoire virtuelle.

La mémoire virtuelle représente en volume la totalité de la place à partager entre les utilisateurs à un instant donné. Un compteur du reste des K virtuels permet de limiter l'allocation en mémoire réelle aussi bien que virtuelle, même s'il reste de la place réelle dès qu'il risque de devenir impossible de faire sortir en mémoire virtuelle tous les travaux de la mémoire réelle pour les remplacer par un des travaux en attente en mémoire virtuelle.
Ceci suppose que la mémoire virtuelle est au moins aussi grande que le plus gros travail autorisé en mémoire réelle (dans ce cas ce seul travail bloque l'accès au système) ; la limite supérieure dépend de



Echanges entre mémoire réelle et virtuelle

la "charge" du système mais elle peut atteindre légitimement le produit du nombre maximum de travaux par la taille maximum autorisée. Dans ce dernier cas, la place non occupée n'est pas perdue mais sert à la mise en attente de programmes réentrants, partageables et réutilisables. Nous verrons d'un autre côté que l'utilisation de tels segments de programme permet de diminuer la taille de la mémoire virtuelle.

2 - Amorçage du processus d'échange

Le processus d'échange s'amorce :

- lorsque l'occupation de la mémoire réelle est telle qu'une expansion n'est pas possible sans réarrangement,

- lorsqu'une allocation a été faite en mémoire virtuelle et qu'il faut faire rentrer le segment créé, pour exécuter la suite d'un ordre d'un utilisateur.

Au premier réarrangement des queues (voir gestion des tours de passage) un appel sera fait, avant de chercher le prochain travail à relancer, à la routine d'échange entre les mémoires. La première fois, il n'y a pas de segment déjà retenu pour sortir, il n'y a rien à acquitter (tâche différée par excellence), on va rechercher un segment ou plutôt un travail dans les queues d'attente, pour rentrer. S'il n'y en a pas et qu'il n'y a pas non plus de demandes d'expansions externes (par sortie puis rentrée sous une taille supérieure), la routine s'achève rapidement. Si par contre une allocation a été faite en mémoire virtuelle, ou si un des travaux a été placé dans la queue spéciale d'attente pour exécution différée d'un ordre, faute de mémoire réelle, le numéro de travail (identique au numéro de segment bas dont aura besoin le décodeur d'ordre à la prochaine occasion) va être sélectionné pour rentrer en mémoire réelle. Si la place existe en un seul trou, d'autres travaux ayant changé depuis la mise en attente de l'ordre par exemple, l'ordre

d'entrée est immédiatement lancé après réservation de mémoire réelle (assurée de réussir) ; le cas particulier où l'image externe n'existe pas (après création d'un segment virtuel fictif), il n'y aura pas de transfert lancé, mais retour immédiat à l'acquittement de transfert d'entrée pour continuer comme après un véritable transfert.

3 - Le système de tassement en mémoire réelle

La première fois, très certainement, le transfert ne pourra être lancé sans une tentative de tassement pour regrouper les trous, auquel cas il faudra peut-être attendre plusieurs interruptions d'horloge, plusieurs passages dans cette routine avant de réussir à bouger le segment situé au-dessus du premier trou. Avant de bouger un segment il faut s'assurer qu'aucune routine de périphérique n'est en train d'y effectuer des entrées-sorties ; le segment pour qui ce cas se produit est forcément bas, (les entrées-sorties sont interdites dans les segments hauts dont la vocation est de contenir du code), le travail, de même numéro que ce segment bas n'est plus apte à être sélectionné pour tourner (il risquerait de relancer d'autres entrées-sorties) et de plus à la fin du traitement du tampon (voir opérateurs programmés) en cours, il n'y aura pas chaînage sur le tampon suivant comme il est fait normalement par les routines d'interruptions d'entrées-sorties. Il y a ainsi bon espoir de déloger le trou le plus bas, en faisant apparaître un autre plus haut et de proche en proche un trou unique assez grand pour faire rentrer ou créer le segment désiré. Par tassement, on peut faire l'économie de la sortie d'un travail pour en rentrer un autre, (déplacer 1 K prend 2,5 ms, sortir 1 K prend 17 ms + 13 ms par K, pendant ce temps l'unité centrale fait autre chose mais le contrôleur de disque et le canal sont immobilisés et ne peuvent servir aux fichiers). Dans d'autres cas, un tassement s'avérerait insuffisant, il va falloir choisir qui sortir, en donnant la priorité à ceux dont la sortie pour expansion externe est indispensable et la plus assurée (contrôle déjà fait de l'absence

d'entrées-sorties en cours), sinon le plus gros travail dont un paramètre, le temps de protection (initialisé à la rentrée en mémoire réelle proportionnellement à la longueur du travail) est réduit à Ø et divers autres paramètres dont la queue dans laquelle il se trouve, va être sélectionné pour sortir en commençant par le segment haut s'il existe et s'il en est le seul utilisateur (segments hauts sont partageables).

4 - Sortie d'un segment : conditions à vérifier au préalable

Une fois sélectionné, son segment haut ou bas, pour sortir ce travail ne sera plus apte à tourner et si des entrées-sorties sont en cours, elles devront s'arrêter à la première occasion. Si, par hasard, c'est l'utilisateur courant, interrompu pour traiter les tâches différées qui a été choisi, le transfert est différé d'un coup d'horloge car un certain nombre de renseignements lui appartenant sont encore éparpillés dans le superviseur et seul le processus de changement de main est habilité à les rassembler et les ranger dans la zone basse du travail en question. Quand enfin, toutes les conditions sont satisfaites pour sortir un segment haut ou bas d'un utilisateur (avec préséance du haut) dont le numéro est bien en évidence pour qu'à l'entrée suivante dans la routine on n'aille pas rechercher un autre candidat à la rentrée, alors la réservation est faite en mémoire virtuelle, un calcul de parité globale entrepris sur tout le segment à sortir, parité qui sera vérifiée à la rentrée de ce même segment, et gardée en mémoire ; l'ordre de sortie est lancé avec éventuellement dissémination de blocs entiers de K dans la mémoire virtuelle si le segment n'a pu sortir en un seul tenant ; l'écriture de fragments sera poursuivie en interruption.

Pendant tout le temps du transfert aucune nouvelle sélection ne sera entreprise ; lorsqu'il sera terminé, on acquittera à la première occasion après vérification qu'il n'y a pas eu d'erreur matérielle (relancement plusieurs fois avant de sortir un diagnostic) et enfin la place occupée par le segment sorti sera libérée en mémoire

réelle avant de procéder à une nouvelle tentative de réservation de place pour la rentrée du segment en attente en mémoire virtuelle.

VII - GESTION DES PROGRAMMES PARTAGES

1 - Segment pur et segment impur

Les programmes sont rangés sous forme de fichiers dans les répertoires des périphériques de stockage (bandes adressables et disques). Un programme est partageable s'il figure dans un répertoire sous un nom (9 caractères) dont les 3 derniers (appelés extension) constituent un code conventionnel réservé dans le système à la désignation de tous les segments partageables.

Un tel programme peut être réclamé implicitement (programme général de bibliothèque) ou explicitement (plusieurs usagers concomitants d'un même programme privé) par un ordre d'exécution (aller chercher, exécuter) ou par un opérateur programmé (lancement de programme, changement de segment haut). Seuls des segments hauts protégés contre l'écriture peuvent être partagés, aucune indication propre à un usager particulier ne peut y figurer, le segment bas de chaque usager est obligatoire et sert de zone impure de rangement comme d'exécution du programme, de zone de rangement pour le superviseur lors des changements de main, de zone d'échange pour les entrées-sorties qui sont définitivement interdites dans les segments hauts même privés et autorisés en écriture afin de libérer ces segments des contraintes qu'impose l'attente de fin d'entrées-sorties dans un segment pour la gestion de la mémoire (tassement, échange avec la mémoire virtuelle).

Les segments hauts ainsi libérés sont les premiers à sortir et éventuellement les seuls lors d'un besoin de place en mémoire réelle, ils sont aussi les derniers à rentrer, des 2 segments d'un utilisateur immobilisé en mémoire virtuelle.

2 - La protection écriture du segment haut

Un segment haut privé peut bénéficier du droit d'écriture sur demande au moyen d'un opérateur programmé, ceci peut être fort utile lors des mises au point de futurs programmes partagés.

3 - Effet sur les sorties en mémoire virtuelle

Pour les performances générales du système, un segment haut a tout intérêt à être protégé en écriture, le superviseur le sachant, peut éviter de faire systématiquement une copie de ce segment en mémoire virtuelle dès l'instant où il y en existe déjà une, la première suffisant et restant valable tant que sa place n'aura pas été nécessaire pour stocker quelque chose d'autre et tant que la copie réelle ne risquera pas de différer de la copie virtuelle (mise en écriture permise ou changement de l'allocation mémoire).

Un segment haut protégé en écriture, même privé, soulage les échanges avec la mémoire virtuelle la première sortie étant suffisante et lors des rentrées ultérieures, l'image virtuelle n'étant pas détruite. Si, de plus, le segment a été reconnu dans le système comme partageable d'après le nom de fichier qu'il portait lors de son premier accès, tout nouvel appel de ce programme se bornera à la création du segment bas correspondant propre au nouvel utilisateur et limitera ainsi les accès du périphérique de stockage de fichiers.

Le nombre de lecteurs du segment partagé est alors incrémenté pour faciliter le contrôle de l'utilisation de ce segment.

4 - Création d'un segment réentrant

Un segment réentrant avec segment haut et segment bas est écrit sous forme de 2 fichiers (un par segment) de même nom avec des extensions différentes en pratique par un ordre de sauvegarde donné au superviseur, sur un périphérique. Le nom donné au fichier contenant le segment haut permet de faire lors d'un rappel du programme, la distinction entre réentrance ou non. Le programme à deux segments sauvé de la mémoire par l'ordre frappé est en général issu d'un programme à un seul segment bas scindé par un opérateur programmé ayant fait apparaître de toutes pièces le segment haut, c'est ainsi que le programme chargeur crée les nouveaux programmes à 2 segments. Le segment réentrant ainsi sauvé pour être compatible avec les ordres de recherche ou d'exécution du superviseur se présente sous forme de fichier, image de la mémoire sans compression de zéros car en général, un tel segment n'a que du code sauf au-delà de la dernière adresse, l'appel d'un tel segment ne nécessite pas d'expansion de code comme pour les segments bas ou les programmes à un seul segment (comme le superviseur).

5 - Appel d'un segment réentrant

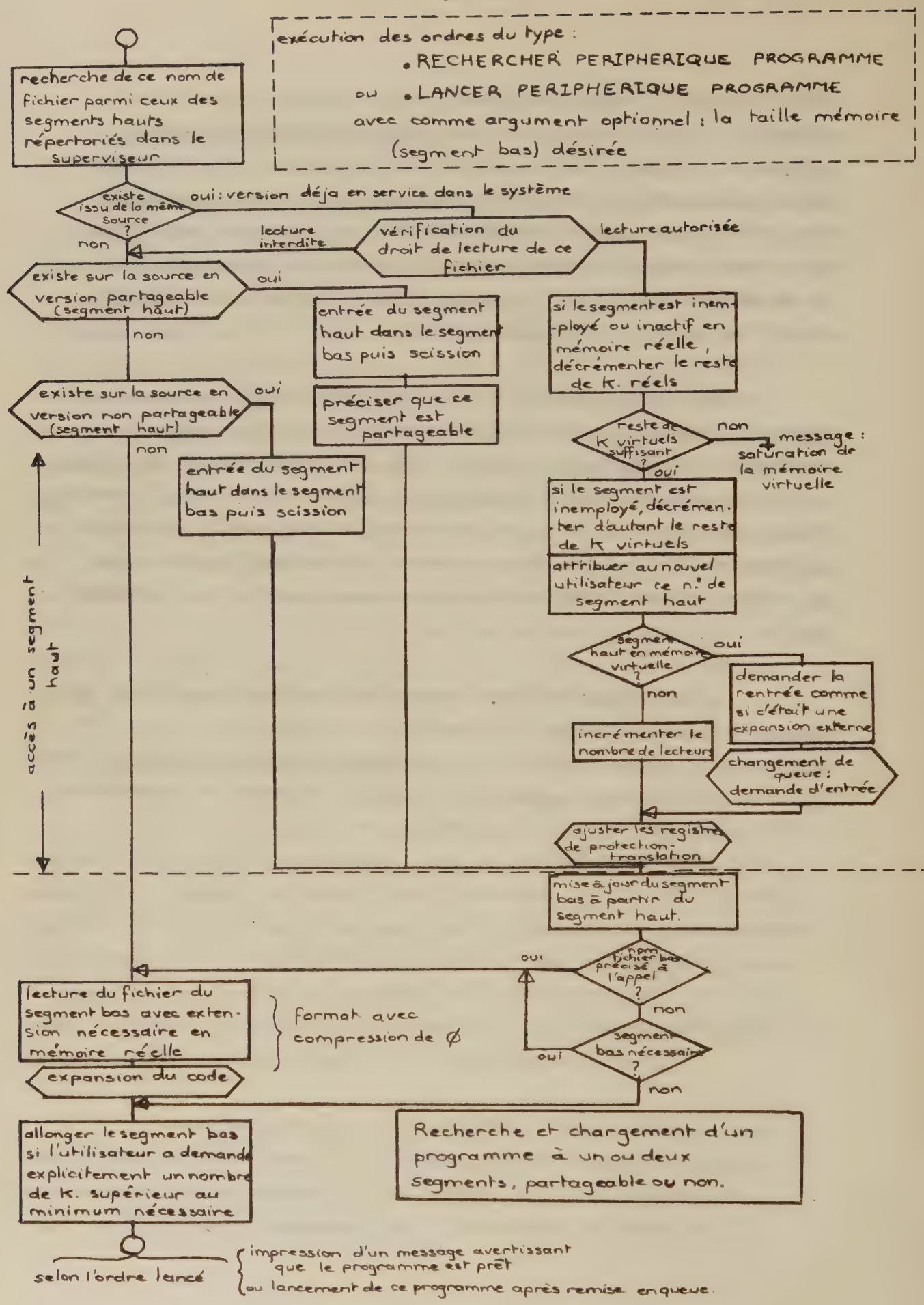
L'accès à un programme réentrant est fait par les mêmes ordres que pour les programmes ordinaires, sur des périphériques munis d'un répertoire. L'accès à un segment haut (réentrant ou non) est également possible par un opérateur programmé issu d'un segment bas. (un programme à 2 segments qui s'est débarrassé de celui du haut peut éventuellement le rappeler plus tard ou un programme utilisateur peut très bien faire appel en cours d'exécution à un segment haut, certainement réentrant et partagé appartenant par exemple à la bibliothèque système).

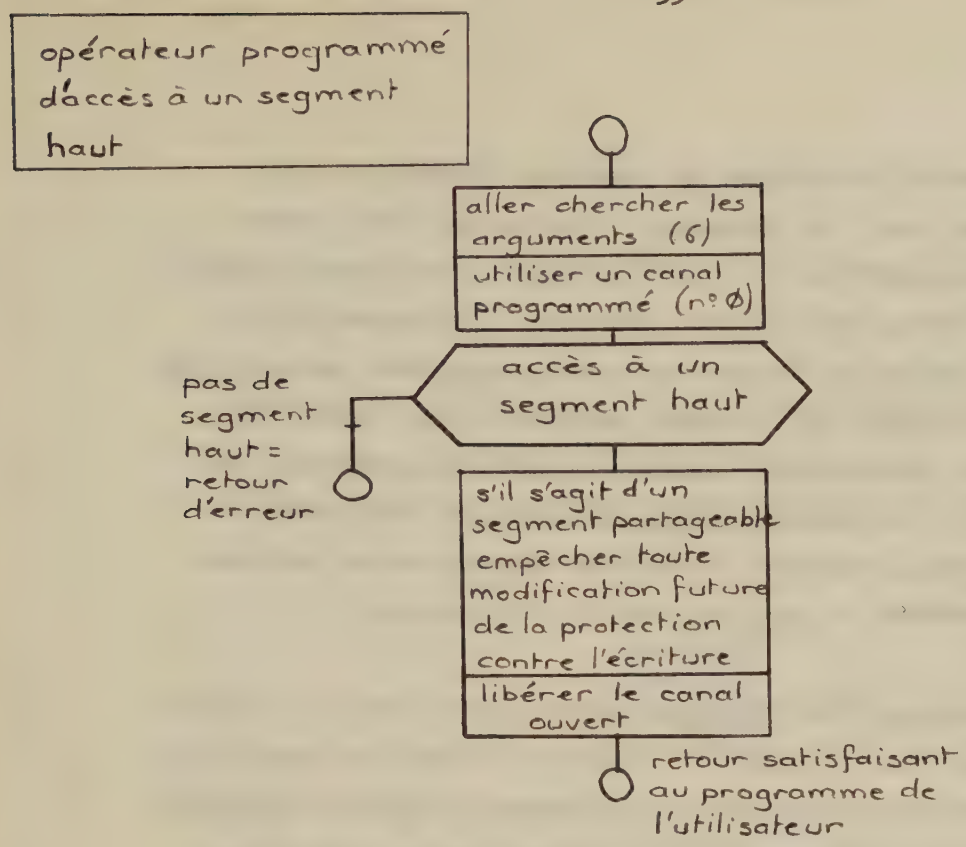
L'ordre des opérations de recherche de programme fait que sur le répertoire du support de fichiers peuvent figurer une version à 2 segments et une version à 1 seul segment bas d'un programme de même nom (6 premiers caractères) mais d'extensions différentes. Si

aucune extension (valable uniquement pour le segment du bas) n'est précisée la version partageable sera prise en priorité en mémoire centrale ou à défaut, sur la source, puis la version haute non partageable après quoi, si l'extension est précisée, le segment bas de la version à un seul segment est chargé (sans que le segment haut disparaisse pour autant) écrasant les positions mises à jour en bas à partir du segment haut (dont l'adresse de départ) permettant de forcer ainsi l'utilisation de la version désirée.

Un segment haut porte comme indication la plus haute adresse chargée de données sauvées dans le segment bas, si cette adresse est inférieure à la plus basse possible (140, car de Ø à 140 les positions servent à communiquer avec le superviseur) aucun segment bas n'a besoin d'être recherché sous un nom de fichier et le chargement du programme est facilité d'autant. La réservation de mémoire pour le segment bas est faite néanmoins au moyen d'un des arguments sauvés dans le segment haut.

L'accès programmé à un segment haut utilise la partie de la routine précédente chargée de rechercher le segment haut, le nom peut être partiel (recherche avec les extensions implicites) ou complet ainsi que d'autres arguments passés par l'opérateur programmé au superviseur.


exécution des ordres du type :
• RECHERCHER PERIPHERIQUE PROGRAMME
ou • LANCER PERIPHERIQUE PROGRAMME
avec comme argument optionnel : la taille mémoire (segment bas) désirée
recherche de ce nom de fichier parmi ceux des segments hauts répertoriés dans le superviseur
existe issu de la même source ?
oui : version déja en service dans le système
non
vérification du droit de lecture de ce fichier
lecture interdite
lecture autorisée
existe sur la source en version partageable (segment haut)
oui
non
entrée du segment haut dans le segment bas puis scission
préciser que ce segment est partageable
existe sur la source en version non partageable (segment haut)
oui
non
entrée du segment haut dans le segment bas puis scission
si le segment est inemployé ou inactif en mémoire réelle, décrémenter le reste de K. réels
reste de K virtuels suffisant ?
non
message : saturation de la mémoire virtuelle
oui
si le segment est inemployé, décrémenter d'autant le reste de K virtuels
attribuer au nouvel utilisateur ce n.° de segment haut
segment haut en mémoire virtuelle ?
oui
demander la rentrée comme si c'était une expansion externe
non
incrémenter le nombre de lecteurs
changement de queue : demande d'entrée
ajuster les registres de protection-translation
accès à un segment haut
mise à jour du segment bas à partir du segment haut.
fichier bas précisé à l'appel ?
oui
non
segment bas nécessaire ?
oui
non
lecture du fichier du segment bas avec extension nécessaire en mémoire réelle
format avec compression de Ø
expansion du code
allonger le segment bas si l'utilisateur a demandé explicitement un nombre de K. supérieur au minimum nécessaire
Recherche et chargement d'un programme à un ou deux segments, partageable ou non.
selon l'ordre lancé
impression d'un message avertissant que le programme est prêt
ou lancement de ce programme après remise en queue.



6.- La protection écriture et les conditions de sa levée

Un certain nombre de précautions sont requises pour prévenir tout accident avec des programmes qui risquent d'être partagés. Un segment partageable comporte une protection à la fois contre les lectures (exécutions) simultanées du programme lui-même par un utilisateur non autorisé et contre l'écriture ou la levée de protection-écriture par un utilisateur autorisé en lecture.

La protection appliquée à un segment est la même que celle concernant l'accès au fichier-source sur un disque public. Pour ces fichiers, on peut préciser si le propriétaire du fichier, quelqu'un de son équipe ou quelqu'un d'autre peut lire ou réécrire (modifier) un fichier donné.

La mise à jour d'une source de programme (fermeture d'un nouveau fichier de même nom), le changement physique de cette source (pile de disque, bande magnétique à répertoire après une nouvelle particularisation de son dérouleur) doivent provoquer l'effacement dans le superviseur, de la source et du nom du ou des programmes partageables ainsi mis à jour ou provenant de la source modifiée (pour forcer le superviseur à utiliser les nouvelles sources ou nouvelles versions en cas de nouvelle demande des segments hauts partageables concernés). Les utilisateurs courants continuent d'utiliser les anciennes versions qui ne sont plus référencées à leur nom dans le superviseur.

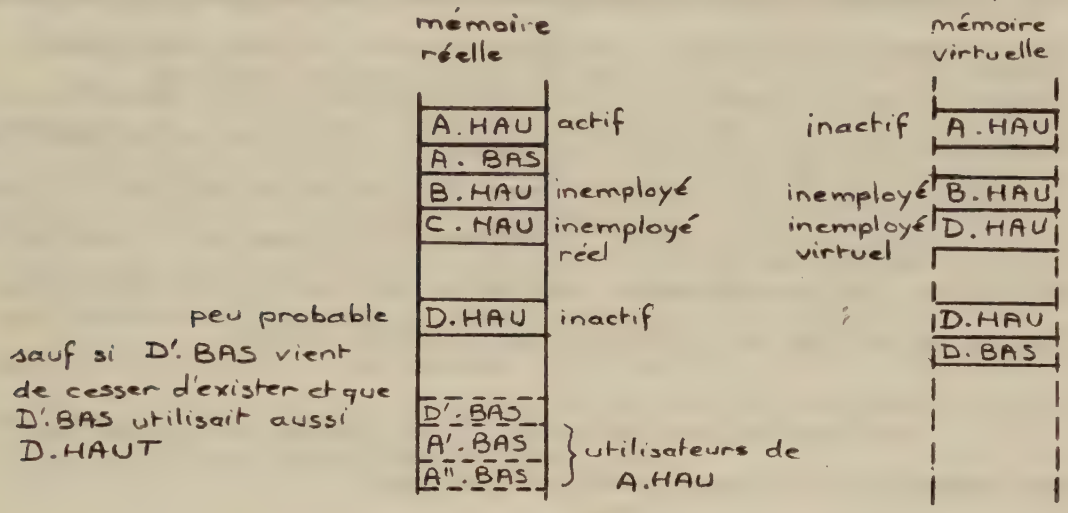
Les bibliothèques, les fichiers peuvent être ainsi mis à jour sans avoir à se soucier des utilisateurs possibles actuels et sans risque de méprise lors des appels futurs : un programme réentrant bénéficie des mêmes avantages qu'un fichier ouvert en lecture sur un support public du genre disque. (voir la gestion des fichiers).

Ces contraintes sont faibles par rapport aux gains de place et de temps réalisés par le partage d'une même partie du code et l'effacement instantané pour récupérer sa place d'un segment haut quand une version existe déjà en mémoire virtuelle pour satisfaire une recopie future.

7 - Problèmes du maintien inactif en mémoire des segments réentrants : avantage de leur survie

La technique consistant à maintenir les emplacements des segments réentrants momentanément inutilisés tout en les considérant comme immédiatement effaçables en cas de besoin de leur place permet une économie de transferts au moment de leur réemploi s'ils n'ont pas été effacés, tout en n'encombrant pas la mémoire (réelle comme virtuelle). Ils sont inemployés quand aucun utilisateur ne les utilise, ils sont inactifs quand un segment bas qui y fait référence se trouve dans la mémoire de type opposé ainsi qu'une image de type contraire.

Images réelles et virtuelles des segments



VIII - GESTION DES TOURS DE PASSAGE

1 - L'attribution du numéro de travail - Liaison avec la console de l'utilisateur

La plupart des ordres donnés au superviseur nécessitent l'attribution d'un numéro de travail, en fonction de l'ordre d'arrivée dans le système, afin d'organiser les tours de passage dans le partage des ressources du système. Il existe en outre un état transitoire pendant lequel ce numéro a été attribué mais où il est impossible de bénéficier d'autre chose que du programme en cours de déroulement dont le rôle est de mettre à jour le livre de bord et de vérifier l'identité (mot de passe, nom) du nouvel utilisateur du système ou, à son départ, d'inscrire les renseignements comptables dans un fichier du système en vue de la facturation ultérieure des services rendus.

Au cours de ces processus, l'utilisateur ne dispose pas, par exemple, de l'ordre d'arrêt de programme.

Le numéro de travail est confondu avec le numéro de segment bas de l'utilisateur dans les tables indexées par les numéros de segments. Les numéros de segments supérieurs au nombre maximum d'utilisateurs prévu à la définition de la configuration (voir construction du superviseur) correspondent à des segments hauts, partageables ou non (actifs, inactifs ou inemployés).

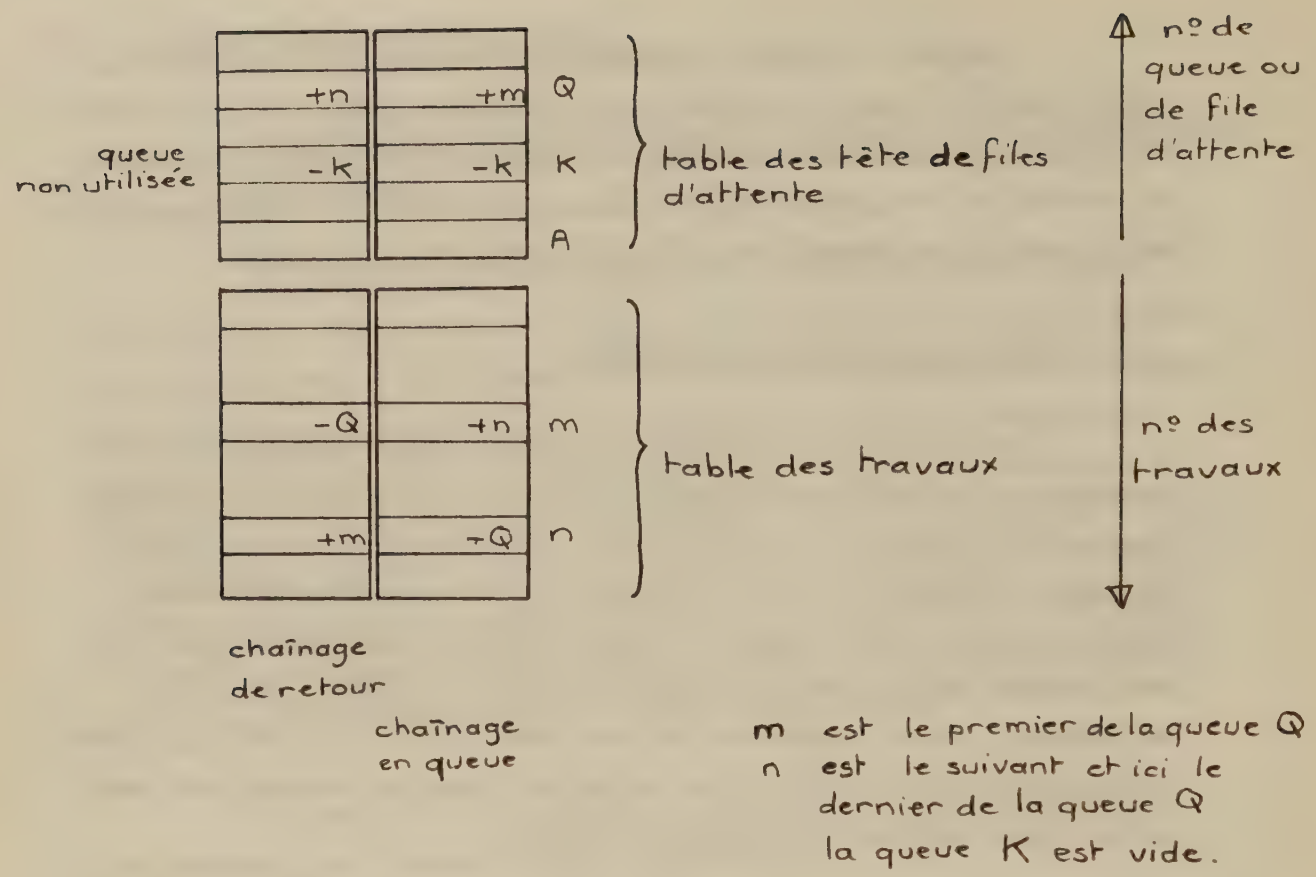
Un utilisateur ne dispose que d'un seul numéro de travail à la fois; comme il ne dispose que d'une console, pour mener plusieurs travaux d'une même console, ceux-ci sont contraints de s'ignorer et de toute façon un seul est à un moment donné attaché à la console, les autres ont pu être abandonnés dans le système, arrêtés ou en cours de traitement avant l'ordre de désaffectation de la console donné au superviseur.

Lorsque la console a abandonné un travail, il est possible de se réinscrire dans le système sous un nouveau numéro ou de rappeler un travail abandonné par un ordre de réaffectation, pour en reprendre le contrôle ou finalement pour le rayer du système et faire comptabiliser la part de ce travail dans l'utilisation des ressources.

2 - Les files d'attente du système

Un travail est caractérisé par un certain nombre de tables indexées par le numéro du travail, certaines de ces tables concernent plus particulièrement sa position dans l'utilisation des ressources partagées que sont le temps de passage et les priorités relatives. L'utilisateur n'a pratiquement aucune prise sur ces indications, il peut arrêter son travail, en lancer un autre ou le même, avoir fait des erreurs graves l'empêchant de continuer mais c'est tout, les indications sont en fait gérées par des algorithmes de priorité sur lesquels chacun individuellement ne peut rien.

A un moment donné, un travail appartient à une classe unique de travaux avec dans cette classe une priorité de position comme dans une file d'attente. Puisque c'est le nombre d'articles qui est constant et non le nombre de classes, les queues d'attente sont avantageusement organisées en listes enchaînées (d'une cellule d'un mot) en avant et en arrière pour en faciliter l'exploration. Tout changement de queue se résume à une manipulation simple de pointeurs, l'insertion peut être faite en tête ou en queue d'une file d'attente selon les manipulations.



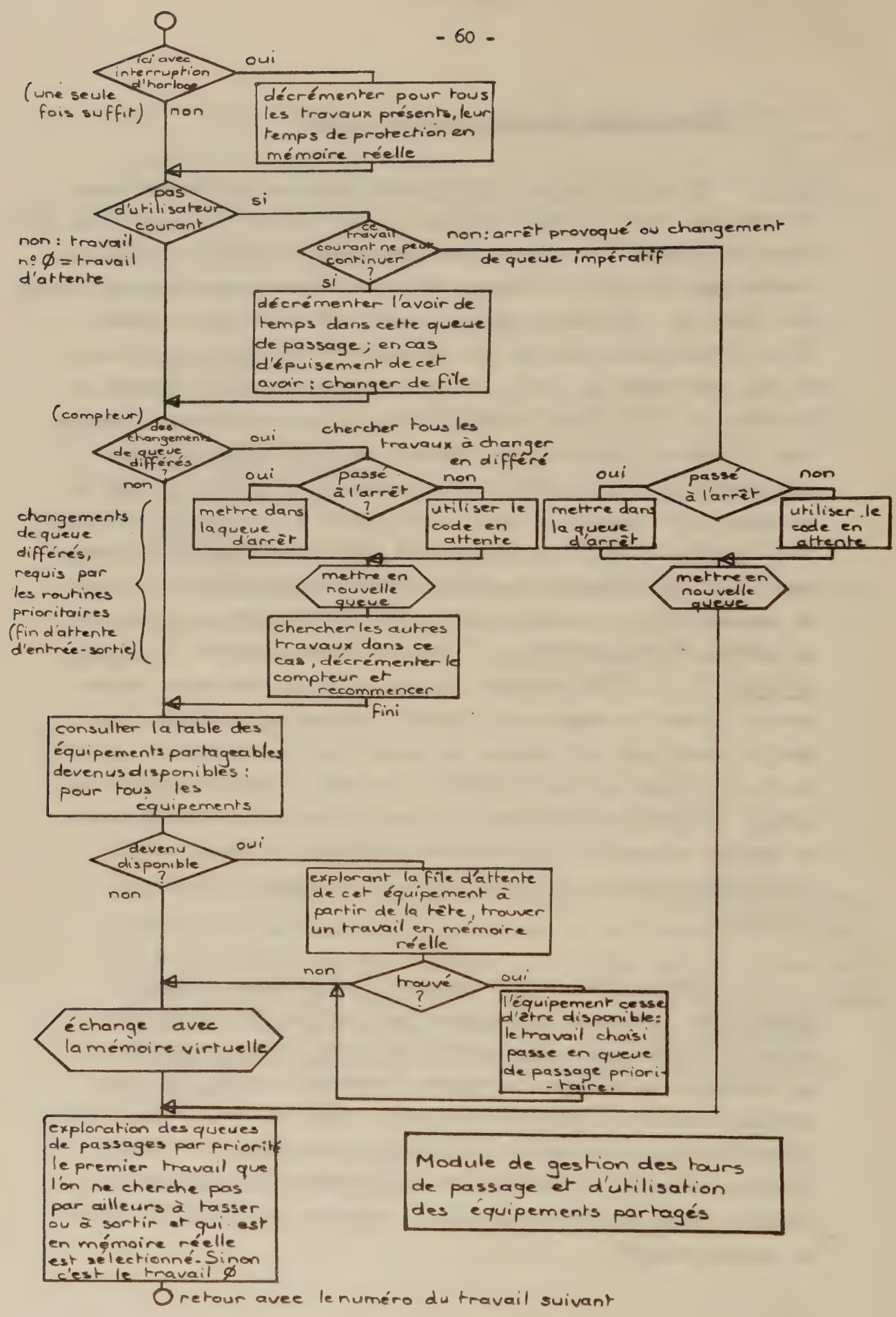
m est le premier de la queue Q
n est le suivant et ici le
dernier de la queue Q
la queue K est vide.

Le nombre des queues et le nombre des travaux peuvent être modifiés indépendamment des algorithmes de modifications des queues avec une économie de place maximum ; un pointeur négatif indique un numéro de queue donc la fin de la file, un pointeur positif un numéro de travail dans les 2 tables. Initialement tous les travaux sont chaînés dans la queue nulle et les autres queues sont bouclées sur elles-mêmes.

3 - Déclenchement des changements de queues

Un mot d'état par travail porte entre autres indications, le temps qui reste à passer dans la queue actuelle s'il s'agit d'une des queues de passage (les autres étant des queues d'attente avant de retourner dans une queue de passage dès que l'obstable empêchant de passer aura été levé : équipement partage immobilisé par un autre travail). Le mot d'état porte aussi un bit indiquant si le travail peut tourner ou non, positionné là pour faciliter les tests et forcer l'entrée ou la sortie de la file des travaux arrêtés, il porte également l'indication d'attente d'exécution de commande, en vue de passer en queue prioritaire pour la rentrée en mémoire réelle, et surtout le code de la queue dans laquelle le travail doit passer à la prochaine interruption d'horloge quand c'est une routine prioritaire (une fin d'entrée-sortie sort de l'attente d'entrée-sortie un travail écarté des queues de passage) qui en a décidé ainsi ou dès que le superviseur décodant un opérateur programmé décide d'attendre, pour poursuivre le travail en cours, la fin d'un transfert (entrée en queue d'attente d'entrée-sortie). Quand la demande est différée (routine prioritaire) c'est que le travail qui doit changer de queue n'est pas le travail courant, la fin de l'interruption prioritaire relance le travail courant ; le travail concerné par la priorité attend son tour et change de queue au premier parcours de la routine de changement de queue sauf si l'utilisateur courant est précisément l'utilisateur Ø en travail d'attente.

Dans ce cas, il y a tout intérêt à relancer le travail concerné par la routine prioritaire (par exemple, il n'a plus de raison d'attendre une entrée-sortie qui vient juste de se terminer) en provoquant un réexamen des queues d'attente, ceci est fait par demande comme pour l'horloge temps réel, d'une interruption au niveau le moins prioritaire sans toutefois positionner l'indicateur d'horloge lui-même de manière à ne pas perturber les quantités temporelles (date, contrôles des délais, temps de protection des travaux en mémoire réelle).



Module de gestion des tours de passage et d'utilisation des équipements partagés

En résumé, les tâches différées non rythmées peuvent être réalisées plus tôt que prévu dans le cas où le superviseur sait que les conditions ont suffisamment changé pour que la poursuite du travail Ø en boucle d'attente soit inutile.

4 - Queues d'attente des périphériques partagés

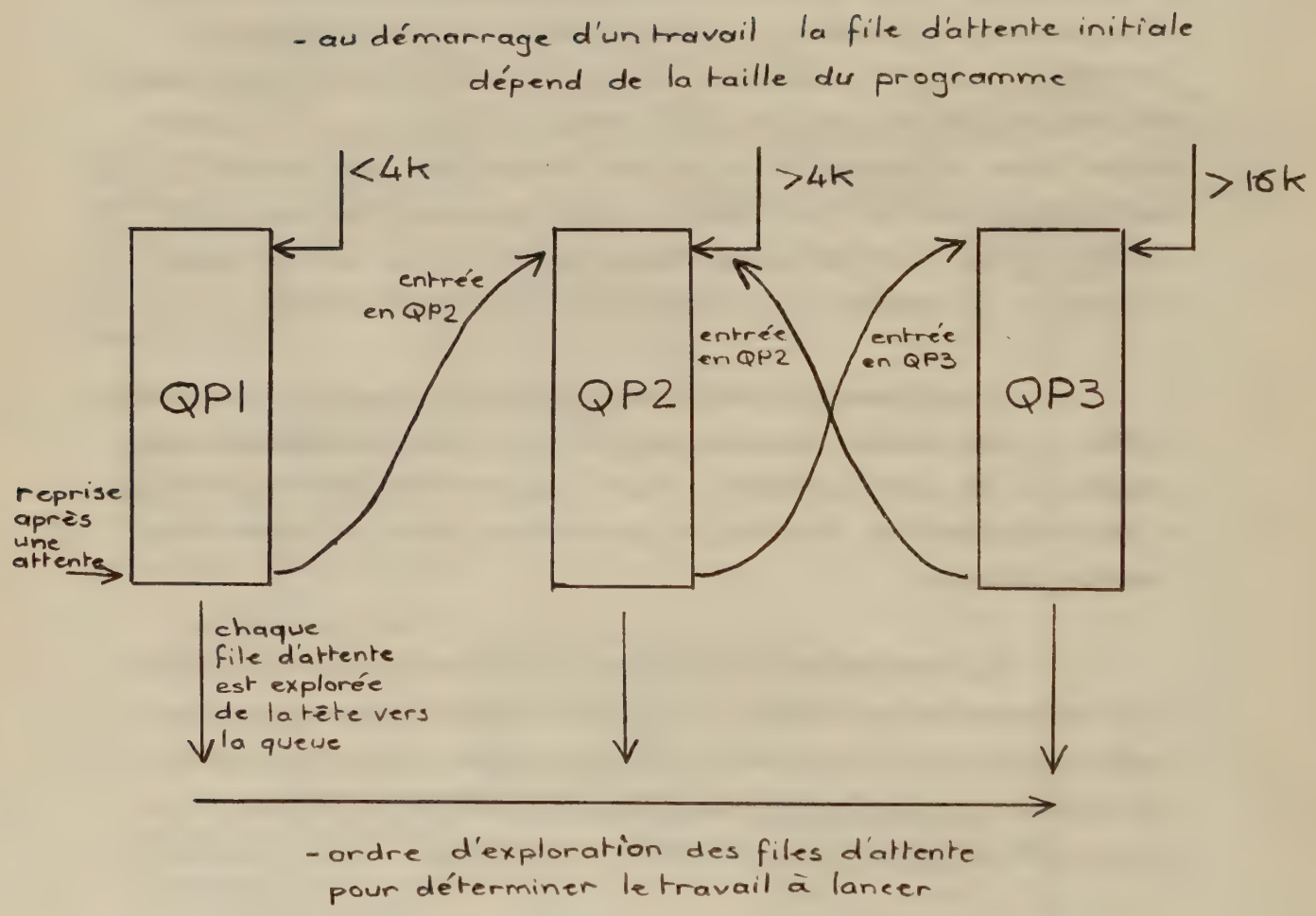
La répartition des équipements partagés est faite facilement par l'existence pour chacun de ces équipements d'une queue d'attente, d'un compteur de demandes (- 1 si disponible, Ø quand utilisé par un travail, positif = nombre de travaux en attente pour cet équipement) et d'un drapeau de disponibilité positionné quand une routine prioritaire libère un équipement et que des attentes existent (compteur > Ø après décrémentation) pour indiquer qu'un changement de queue peut avoir lieu pour profiter de cet équipement et entrer dans une queue de passage le premier de la file d'attente de cet équipement.

Le programme de gestion de ces files d'attente a tout intérêt à être le plus général possible quelles que soient les quantités manipulées, il est ainsi possible de faire face avec les mêmes algorithmes à des configurations diverses, en particulier, quant au nombre d'équipements partagés.

Ces renseignements sont dans des tables faciles à modifier à la construction du système, des tables existent également qui servent de guide pour l'exploration des différentes files d'attente en vue des sorties en mémoire virtuelle, des rentrées en mémoire réelle et du tour de passage dans les queues de passage.

5 - Les 3 queues de passage, priorités et évolution

Il existe 3 queues de passage entre lesquelles les travaux évoluent :



QP1 est la file des travaux reprenant le droit de travailler après une attente (entrée-sortie satisfaite, besoin d'un périphérique partageable occupé). Les travaux y entrent dans ce cas en tête pour avoir le contrôle tout de suite, toutefois, ils n'y disposent que d'un avoir de temps faible.

QP2 est la file des travaux actuellement les plus prioritaires après QP1.

QP3 fait en quelque sorte suite à QP2, le travail qui était le plus prioritaire en QP2 et qui a tourné pendant tout son avoir de temps vient séjourner en queue de QP3 avec peu de chance de tourner maintenant tant que les suivants ne l'auront pas poussé en tête de QP3 puis en QP2.

A l'entrée, dans une file d'attente, un certain temps de travail est alloué (0,5 seconde en QP1, 2 secondes en QP2 et QP3) au bout duquel on passera de QP1 en queue de QP2 et cycliquement dans QP2, QP3.

L'entrée dans le cycle à partir de l'état arrêté est fait en fin de chaque queue en fonction de la taille du travail, les petits travaux étant favorisés car sans doute plus rapides et surtout beaucoup plus faciles à gérer en mémoire. Un travail qui démarre prioritairement en QP1 a de grandes chances de passer puisqu'il sera le premier sélectionné à l'exploration des files pour déterminer le travail à relancer. Les seules raisons pour que le travail le plus prioritaire ne soit pas retenu pour être relancé est son absence de la mémoire réelle (qu'il soit déjà en mémoire virtuelle, que le transfert soit en cours vers cette mémoire ou de retour vers la mémoire réelle) ou que le système attende une fin d'entrée-sortie pour procéder à un tassement par déplacement du segment bas de ce travail. Si tous les travaux plus prioritaires sont ainsi immobilisés un travail peut être sélectionné pour tourner jusque dans la queue QP3.

Lorsque le travail a épuisé son quota dans QP1, il passe en fin de QP2 où, si QP1 est vide, il aura encore la priorité jusqu'à ce qu'il se retrouve en queue de QP3. Alors QP2 a tendance à se vider en remplissant QP3 et lorsque QP2 est vide ou ne contient que des travaux immobilisés, le premier de QP3 tourne et à l'épuisement de son quota passe en QP2 d'où il rejoindra QP3 pour favoriser son suivant. Si tous les travaux ne font que des calculs, QP1 est vide (pas d'attente) ; si de plus la mémoire arrive à contenir tous les travaux, il n'y a pas d'immobilisation pour la gestion, QP2 contient un seul travail qui tourne tout son quota puis passe en queue de QP3, le premier de cette dernière passant ensuite le temps de son quota en QP3, puis le temps de son quota en QP2, les changements sont rares (plusieurs secondes) ce qui économise du temps de gestion pour le superviseur tout en maintenant une bonne répartition du temps de calcul. En fait, les travaux sortent souvent de leur file d'attente QP1, QP2 ou QP3 pour aller dans les queues d'attente, donnant ainsi la main aux plus favorisés qui suivent. A leur retour, ils auront le court bénéfice de repartir en tête. Ceci est surtout destiné à relancer en priorité d'éventuelles opérations d'entrée-sortie et en particulier de bénéficier de la libération d'un équipement partageable dont l'usage a été attendu et dont il faut éviter à tout prix de faire un goulot d'étranglement. Ceci est d'autant plus vrai que cette attente de bénéfice d'un équipement partageable est destinée à relancer une routine du superviseur (en décodage d'opérateur programmé), qui conduit en général rapidement à une remise en attente de fin d'entrée-sortie ; le séjour dans QP1 a toute chance de ne jamais durer jusqu'à l'épuisement du quota, voire même jusqu'à la prochaine interruption d'horloge.

Les tâches pressées et hachées du superviseur interrompent ainsi peu de temps les autres travaux en circulation dans les files QP2 et QP3. Une telle politique vise à ne pas pénaliser les travaux effectuant beaucoup d'entrées-sorties par rapport aux travaux de calcul sur une machine dont le superviseur doit assurer la majeure partie des contrôles et des transferts faute de canaux autonomes suffisamment évolués.

6 - Algorithme de recherche des travaux à sortir de la mémoire réelle ou à rentrer en mémoire réelle

Le même algorithme d'exploration des files d'attente mais fonctionnant en général à l'envers est utilisé pour rechercher les travaux à sortir en mémoire virtuelle afin d'effectuer la rentrée d'un travail immobilisé sur cette mémoire, pour satisfaire le partage de la place. Cet algorithme cherche en priorité, à faire sortir les travaux arrêtés, ceux qui ont demandé à dormir par opérateur programmé ou qui sont en attente d'un périphérique ou d'un contrôleur de périphérique partageable, à l'exclusion peut-être du premier de la file, susceptible d'être appelé à son tour à bénéficier de cet organe. Ces files sont explorées à partir de la queue de façon à ne pas immobiliser, autant que faire se peut, les travaux les plus susceptibles de profiter des ressources les plus coûteuses du système. QP3, QP2 et QP1 sont également explorées, une seule file est épargnée, c'est celle d'attente de fin d'entrée-sortie, les travaux de cette file étant forcément immobilisés en mémoire réelle. La seule exception qui motive l'existence d'une queue d'attente spéciale, est l'attente d'entrée-sortie sur une console d'utilisateur puisque cette console possède un tampon dans le superviseur pour le double usage des modes programme et superviseur et parce qu'une telle attente, en entrée, risque de durer longtemps : elle dépend du rythme de l'utilisateur de la console et il ne pourrait être justifié d'immobiliser un gros travail en mémoire réelle plusieurs minutes en attendant qu'une ligne de texte soit définitivement frappée.

L'attente d'entrée-sortie sur une console n'est pas un obstacle au transfert en mémoire virtuelle d'un travail.

La recherche du premier travail à rentrer, qui précède celle du premier travail à sortir puisque c'en est la motivation, est faite par exploration de toutes les files d'attente dans un ordre favorisant les premiers des queues de passage (QP1, 2 et 3) et les travaux en attente d'exécution d'ordre.

IX - LE DECODEUR DES OPERATEURS PROGRAMMES

1 - Définition et signification des opérateurs programmés

Nous avons vu comment un utilisateur, frappant sur sa console des caractères assemblés en lignes pour constituer des ordres au superviseur pouvait commander l'appel d'un programme dans une certaine zone mémoire et le lancement de ce programme. Ce travail utilise des ressources de place et de temps de travail que le superviseur lui alloue en fonction de l'état général et des besoins des autres utilisateurs éventuels du système. Un tel programme lancé en mode utilisateur n'est pas autorisé à tenter des instructions d'entrée-sortie pour communiquer avec l'extérieur, par exemple, avec la console de son opérateur.
Des instructions-machine d'un certain code permettent par l'intermédiaire du superviseur d'établir ces communications. Ces instructions sont automatiquement rangées à une certaine adresse (40 absolue) et une autre instruction (41 absolue) automatiquement exécutée dans le superviseur, en sauvant le compteur programme et les indicateurs associés ; elle conduit à l'entrée du décodeur des opérateurs programmés.

Les opérateurs programmés réalisent les fonctions courantes dont le programme de l'utilisateur peut avoir besoin :

- la simulation des instructions non câblées pour les systèmes ne disposant pas de l'extension arithmétique (nombres flottants et manipulation de champs, de caractères),

- la communication au programme de certaines données du système : heure, date, nom du système, numéro de travail, caractéristiques physiques d'un équipement,

- la modification des allocations en ressources :
  - . la taille mémoire, le segment haut, partagé ou non,
  - . l'appel d'un autre programme en remplacement de celui-ci,

. la réservation d'équipement et son abandon,
. l'accès au système (programmes de tenue du livre de bord lors du régime transitoire où les ordres sont restreints),

- les transferts d'entrée et de sortie, ces ordres sont les plus couramment utilisés.

Deux routines d'exécution d'opérateurs programmés ont été décrites plus haut pour la gestion de mémoire et pour l'accès à un segment haut éventuellement réentrant. Dans ce qui va suivre, les opérations d'entrée-sortie vont être analysées dans le cas le plus général en particulier quand elles concernent une structure de fichiers sur une série d'équipements publics en insistant sur le symbolisme et les possibilités de partage d'un tel système.

2 - Opérateurs formalisés d'entrée-sortie

L'exécution des opérateurs programmés d'entrée-sortie sera d'autant plus facile et les résultats d'autant plus sûrs que les opérateurs seront mieux formalisés.

La formalisation adoptée est celle de la procédure d'accès par un canal d'échange à un fichier précis, à un enregistrement situé dans ce fichier. Les étapes de cette procédure sont :

- l'initialisation de l'équipement (connexion symbolique de l'équipement au programme)

  au moyen . du nom de l'équipement
  . du mode de travail lors des transferts
  . de l'adresse des pointeurs à utiliser

- l'initialisation d'un fichier de cet équipement
  . en lecture ou en écriture
  . nom du fichier
  . numéro de répertoire s'il y a plusieurs répertoires de fichier sur l'équipement

- l'ordre d'entrée ou de sortie d'un enregistrement
  - . quantité précise à une adresse précise
  - . quantité implicite à une adresse implicite retrouvable par les pointeurs indiqués plus haut

- l'ordre de test de l'état du transfert
  - . indicateurs d'erreur et de fin de fichier, utilisables par le programme

- la fermeture en entrée ou en sortie du fichier ouvert provoquant la fin des transferts implicites non effectués en sortie, l'écriture des marques de fin de fichier, la mise à jour des répertoires.
  L'ouverture ou l'initialisation d'un fichier provoquent autant que possible la fermeture du fichier précédent ouvert dans le même sens sur le même équipement.

- la modification des références d'un fichier dans son répertoire (nom, date, modalités d'accès)

- l'abandon de l'équipement ayant pour effet la réécriture physique des répertoires modifiés et la remise en disponibilité, si cet équipement n'a pas été réservé par un ordre spécial frappé à la console de l'utilisateur et ainsi soustrait des ressources communes partageables.

Dans le système étudié, tous ces ordres existent et s'appliquent à tous les équipements réels ou virtuels de l'installation.

Ces opérateurs programmés disposent dans le code d'instruction du champ normalement réservé au registre (second opérande) pour les instructions câblées, pour désigner en plus le numéro logique, dans le programme, de l'équipement en question. Ainsi 16.équipements peuvent être utilisés en même temps par un seul utilisateur dans son programme. Le superviseur contrôle la logique de la suite des ordres lancés pour un même numéro d'équipement au moyen d'une table de 16. mots située en partie basse de la zone mémoire de l'utilisateur. Cette table contient dès

l'opération d'initialisation de l'équipement l'adresse de la table d'état dans le superviseur, de l'équipement réel connecté à ce numéro logique d'équipement. Puisque cette table est située dans la zone adressable de l'utilisateur, pour éviter son altération ou sa destruction, copie en est faite pour le décodeur d'opérateurs programmés avant le lancement d'un nouveau travail, de son segment bas vers le superviseur et avant le changement d'utilisateur du superviseur vers le segment bas du travail mis en attente, version à jour non modifiable puiqu'en attente.

Cette procédure si elle a l'inconvénient de consommer du temps (2,5 µs par mot transféré) est conforme à la tendance générale qui consiste à placer le plus d'information possible dans les segments bas plutôt que dans le superviseur car ceux-là peuvent être transférés en mémoire virtuelle.

3 - Appellation symbolique des périphériques physiques

La formalisation des instructions d'échange trouve tout son intérêt dans la souplesse d'utilisation offerte si on lui adjoint la possibilité d'appellation symbolique des équipements. Celle-ci est facultative, elle peut donc sans encombre être instituée prioritaire sur l'appellation normale interne au système (IMP désigne l'imprimante, BANDE2 le dérouleur de bande n° 2). L'utilisateur peut donner un ordre de réservation et d'appellation symbolique, au superviseur, depuis sa console, par exemple :

. APPELER DISQUE IMPRIMANTE ou IMP
. APPELER BANDE4 CHOSE

et utiliser dans son programme lors des initialisations les noms IMP et CHOSE faute de disposer au moment de l'exécution des véritables équipements :

soit qu'ils ne soient pas disponibles (IMP en panne ou utilisée par un autre travail),

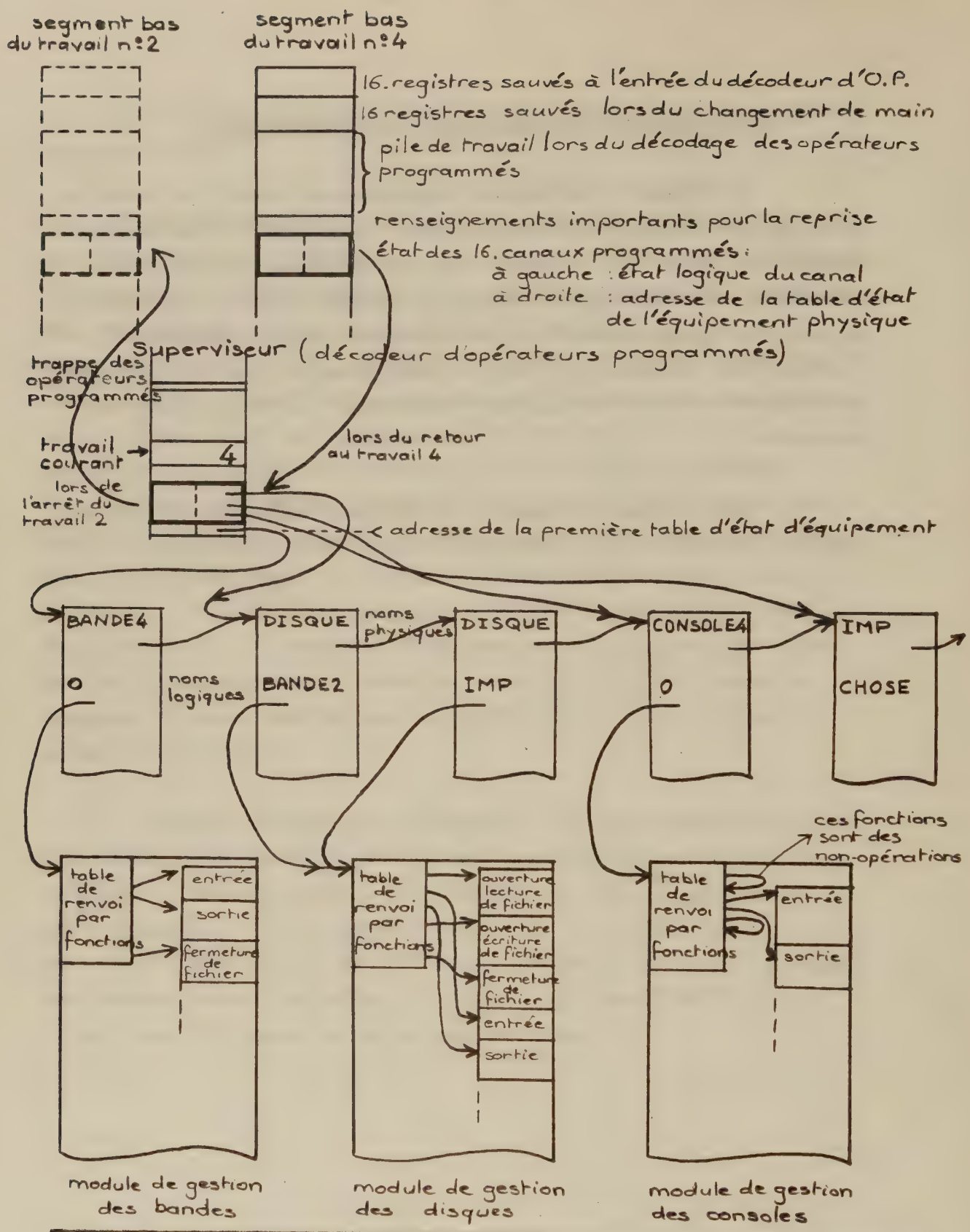
soit qu'ils n'existent pas dans les ressources du système (CHOSE n'est pas un nom interne, DISQUE est un nom interne). Des messages d'erreurs indiquent bien sûr les impossibilités logiques graves pour le déroulement du programme : après initalisation et ouverture de fichier, un ordre d'écriture sur LDC, le lecteur de cartes, n'est pas possible. Par contre, le décodeur et les routines particulières à chaque équipement ignorent les ordres physiquement inexécutables (par suite des appellations symboliques) n'altèrent pas les échanges de données (rebobinage pour le DISQUE remplaçant une certaine bande magnétique). Cette possibilité permet une plus grande compatibilité des programmes entre des installations de ressources et de niveaux d'exploitation différents, ceci permet de modifier ou de décider au dernier moment des supports réels d'information nécessaires à un programme en fonction des disponibilités du moment et du besoin de l'utilisateur du programme (volume, temps de sortie des résultats) sans modification du texte du programme.

Même si sur certains équipements, les opérateurs programmés d'ouverture et de fermeture de fichiers sont inopérants (IMP, CONSOLE, LDCartes) et traités comme des non-opérations, pour la généralité des programmes, il est recommandé de prévoir la procédure complète pour pouvoir répondre au cas où c'est un disque ou une bande adressable avec répertoire qui remplacera par appellation symbolique préalable la véritable IMPrimante ou le lecteur de cartes.

4 - Organisation du décodage pour les entrées-sorties

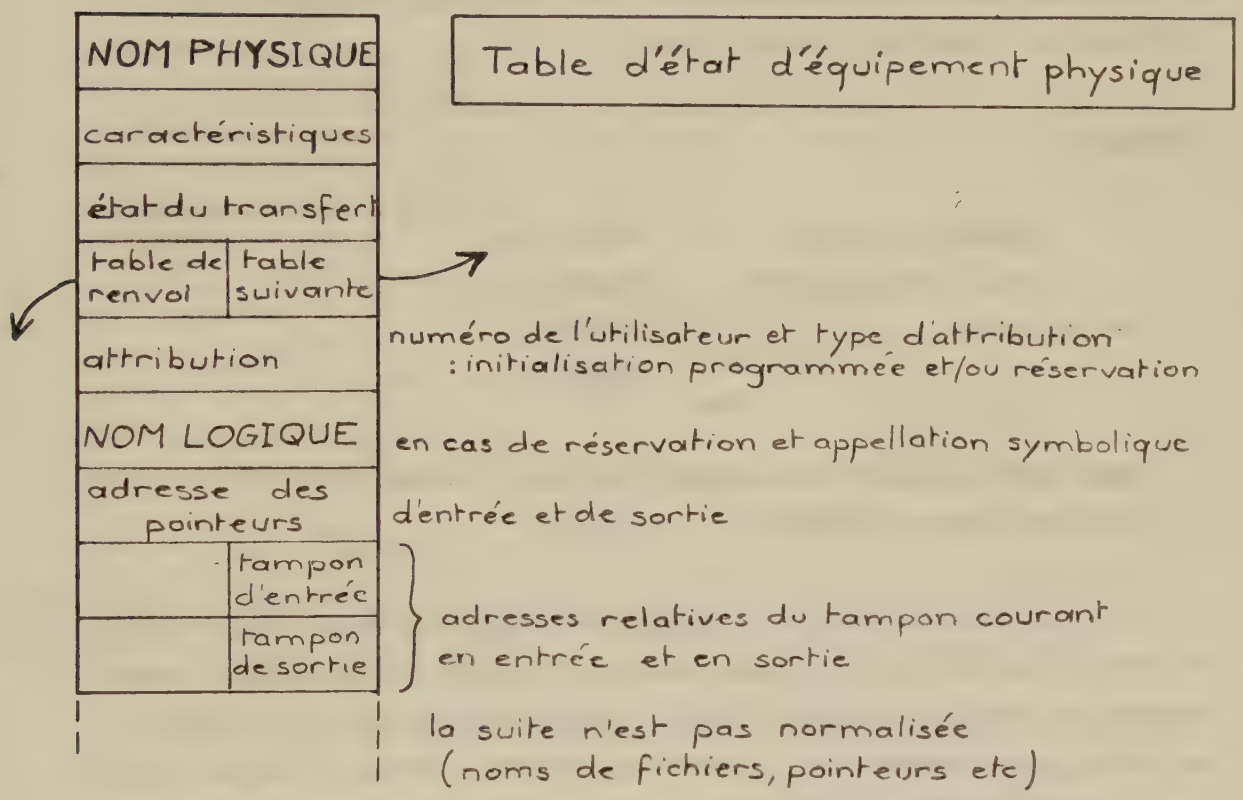
Le décodeur d'opérateurs programmés est chargé de la vérification de la logique des opérateurs quels que soient les équipements physiquement concernés. Il ignore les organes réels effectuant les transferts et ne fait appel à eux qu'en dernier ressort. L'avantage est une plus grande généralité de ce décodeur et son indépendance à l'égard des ressources du système (configuration) sauf quelques cas particuliers comme la console de l'utilisateur qui fait quelquefois exception.

Cette structure demande aux routines spécifiques des équipements de présenter face au décodeur un mode d'accès normalisé.



Liaisons logiques pour les entrées-sorties dans le superviseur

Les routines spécifiques ne sont liées au décodeur logique que par l'intermédiaire des tables d'état d'équipement montées en une seule liste enchaînée, à raison d'une table par équipement, individuellement initialisable par opérateur programmé. Cette table possède pour les premiers mots un format imposé, les mots suivants et la longueur de la table dépendent des équipements, chaque module spécifique de gestion utilisant le reste de la table à son gré.



La table d'état est trouvée lors de l'initialisation de l'équipement où recherche est faite du nom logique parmi les équipements réservés par l'utilisateur (priorité de fait) puis à défaut, du nom physique d'équipement dans la liste enchaînée en prenant soin de n'initialiser que des équipements non réservés ou déjà initialisés par d'autres utilisateurs ; dans certains cas une queue d'attente peut exister (bande système non partageable). L'adresse de la table initialisée est conservée dans la zone utilisateur, dans une table indexée par le numéro logique Ø à 17 de l'équipement, seul utilisé par les opérateurs suivants pour indiquer lequel des 16. équipements possibles est concerné.

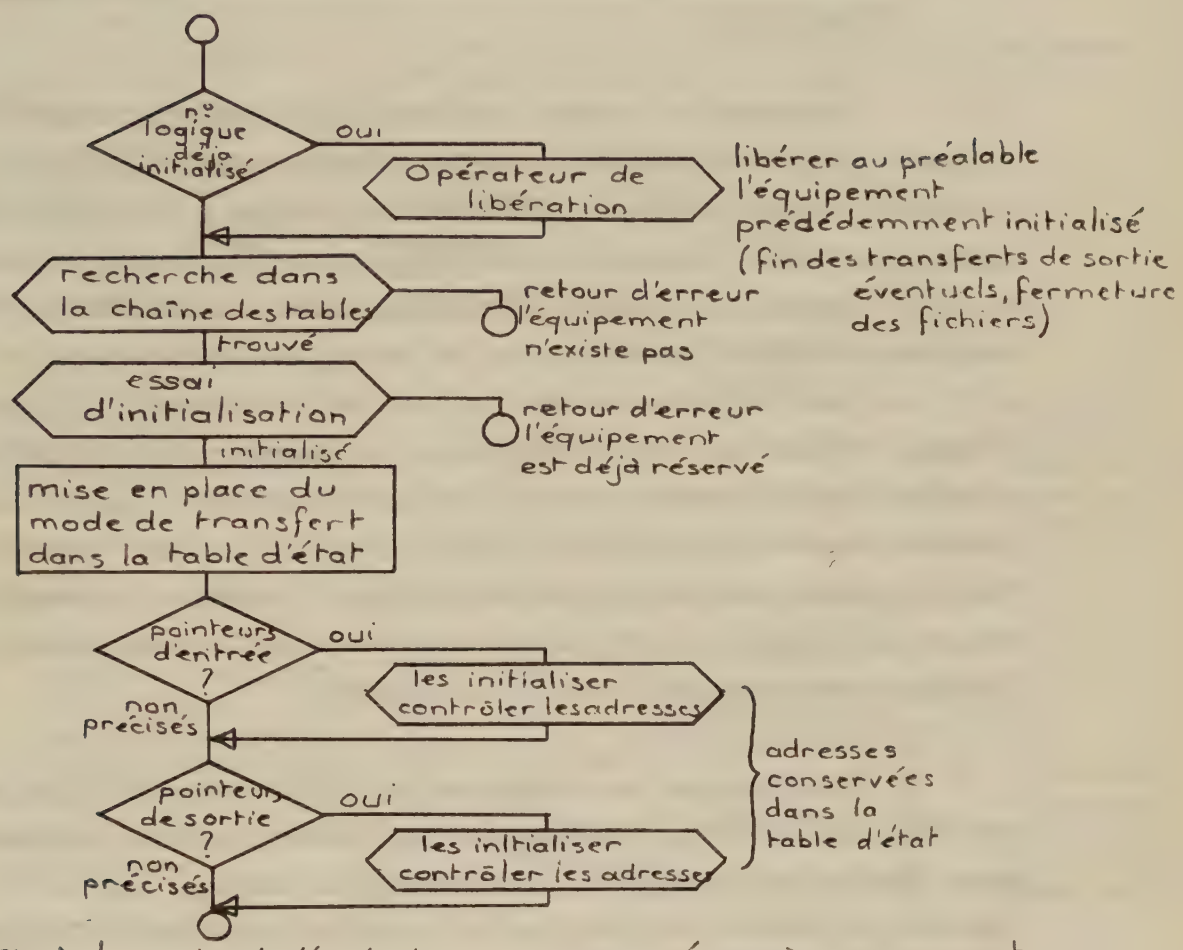
Exemple :

premier opérateur	INIT 4,---
avec comme argument	'BANDE6'
par la suite	SORTIR 4,---
	FERMER 4,---

sans autre argument concernent forcément l'équipement qui a répondu à l'appellation BANDE6

La table d'état de l'équipement porte l'adresse de la table de renvoi (située dans le module chargé de gérer cet équipement) permettant au décodeur un accès facile, par montée de pile à l'adresse de cette table indexée par la fonction désirée (lecture directe, lecture avec tampon, ouverture de fichier, etc...), au sous-module réalisant cette fonction. Certains sous-modules peuvent ne pas exister quand la fonction n'a pas de sens pour l'équipement physique considéré, l'adresse de branchement est selon les cas celle d'une non-opération avec redescente de pile ou bien une routine d'impression de message d'erreur.

5 - Opérateur programmé d'initialisation



retour OK à la suite de l'opérateur programmé après rangement de l'adresse de la table d'état et l'état du canal programmé

Les arguments passés sont :
- le mode (binaire, caractères, ...)
- le nom du périphérique (1 à 6 caractères)
- l'adresse de mise en place des pointeurs d'entrée et de sortie.

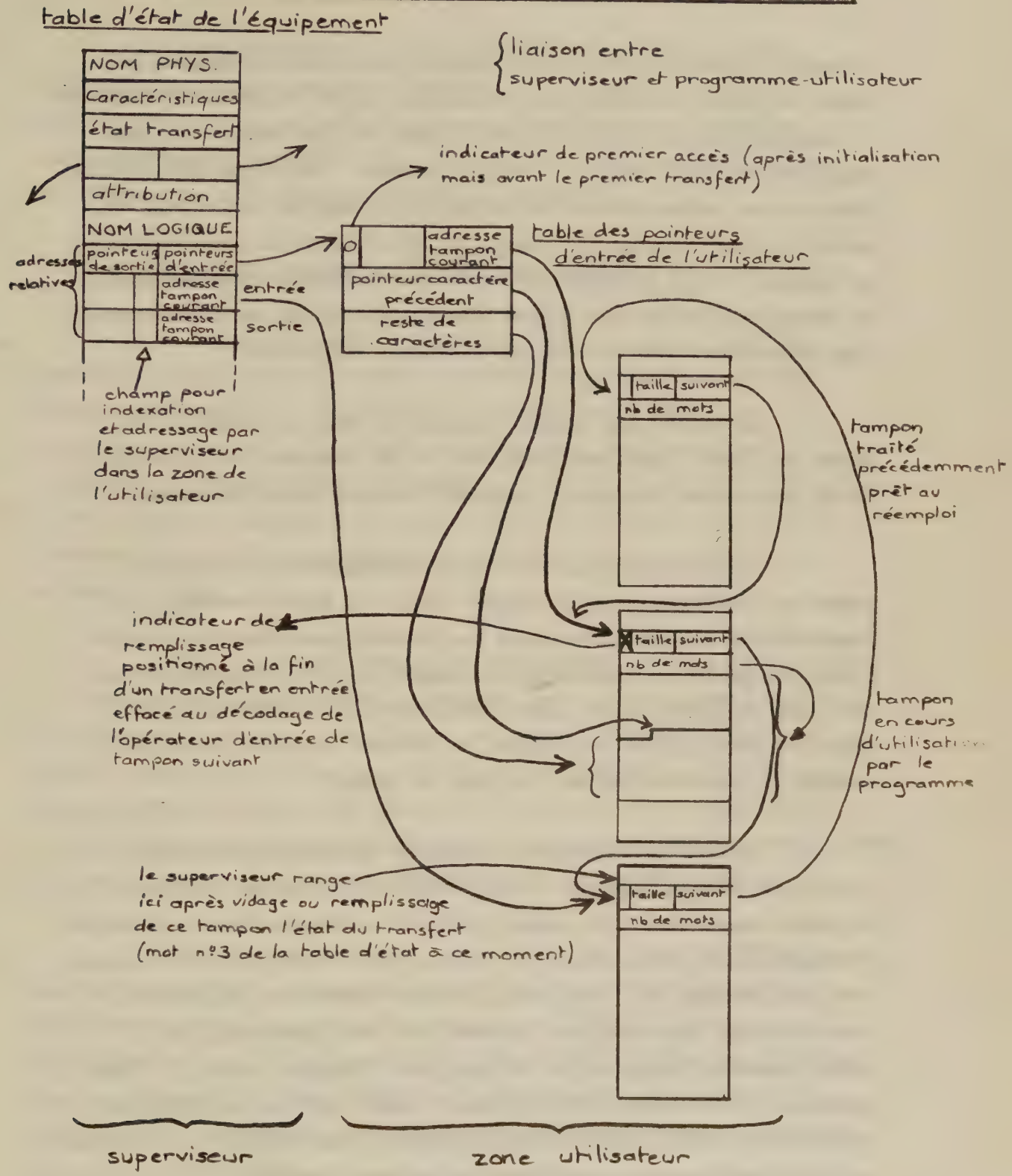
6 - Demande de construction de tampons accessibles par les pointeurs précisés à l'initialisation - Accès séquentiel aux caractères d'un fichier

La demande de tampons d'entrée ou de sortie ne demande outre le numéro logique d'équipement que le nombre de tampons désirés. L'adresse d'implantation est implicite et indiquée à une position du segment bas censée contenir la première adresse libre dans la zone de l'utilisateur. Cette quantité est d'ailleurs initialisée par un opérateur de remise à zéro logique de toutes les entrées-sorties qu'il est bon d'utiliser en tête de tous les programmes. La valeur initiale utilisée est celle fournie par le chargeur avec l'adresse de départ, de la première position où aucun chargement n'a été fait : première adresse libre, elle est mise à jour par le superviseur au fur et à mesure que des tampons sont bâtis, l'expansion de mémoire est de plus automatique en cas de dépassement de la zone allouée à l'utilisateur, lequel n'a pas à s'en soucier.

L'utilisateur est néanmoins libre de faire construire par son programme lui-même, les tampons qu'il désire utiliser, il peut aussi bien ne pas demander de construction de tampons, sauf s'il travaille en mode direct avec mot de contrôle, 2 tampons lui seront bâtis à son premier ordre d'entrée ou de sortie si ces tampons n'existent pas. Ce genre d'information est mémorisé dans l'état du canal programmé et du seul ressort du décodeur d'opérateurs programmés.

Les tampons ainsi construits (d'une longueur indiquée dans la table d'état de l'équipement parmi les caractéristiques) sont chaînés entre eux pour former une boucle. Les routines du superviseur, propres à chaque genre de périphérique font accès directement à ces tampons au moyen du pointeur dans la table d'état sur le tampon courant qui pointe sur le suivant (etc...). Aucune restriction n'existe sur la position des tampons dans le segment bas du programme, ce qui nécessite l'immobilisation de tout le segment en mémoire réelle

Structure des tampons de lecture ou d'écriture



Toutes les adresses sont relatives à la zone de l'utilisateur. Le superviseur indexant sur un registre qui lui tient lieu de registre de base, les déplacements mémorisés dans la table d'état ou lus dans la zone de l'utilisateur.
Cette méthode facilite la translation des programmes en mémoire réelle.

lorsqu'un transfert est en cours malgré peut être les besoins de place des autres utilisateurs. Remarquons qu'en voulant trop favoriser le taux d'échange, ce à quoi tend l'usage des tampons pour le bon usage des périphériques, on risque de grever l'équilibre dans l'utilisation des autres ressources partagées du système aux dépens de petits travaux de calcul en faveur de gros travaux de manipulations de données ; c'est précisément le rôle de la répartition des tours de passage de lutter contre tout accaparement d'une des ressources sans entrer dans les motivations de chaque utilisateur.

La taille des tampons indiquée en tête de chacun d'eux permet la tenue à jour des pointeurs et des compteurs de caractères (ou de mots selon les modes) associés à chaque chaîne de tampon.

L'utilisation habituelle et recommandée des tampons permet de s'affranchir de la nature réelle du support externe, la compatibilité logique entre tous les organes peut être ainsi satisfaite sans autre contrainte de programmation que la décrémentation avec test du compteur des caractères fournis en entrée ou encore libres en sortie avant d'utiliser le pointeur de champ fourni pour accéder au caractère suivant ou pour le ranger.

Ce mode de travail est celui de la consultation et de l'écriture séquentielles de fichier telles qu'elles se réalisent sur bande magnétique : un tampon contient un enregistrement, l'unité d'information peut être selon le mode le mot de 36 bits ou le caractère (en pratique de 7 bits) mais éventuellement de 1 à 36 bits. Pour certaines applications, le mode peut être synchrone, sur simple demande, chaque ordre de transfert d'enregistrement (1 tampon) provoquant un transfert physique pour lequel on a pu au préalable préciser un numéro logique d'enregistrement (sur disque) ; la lecture ou l'écriture ne sont plus séquentielles mais indexées.

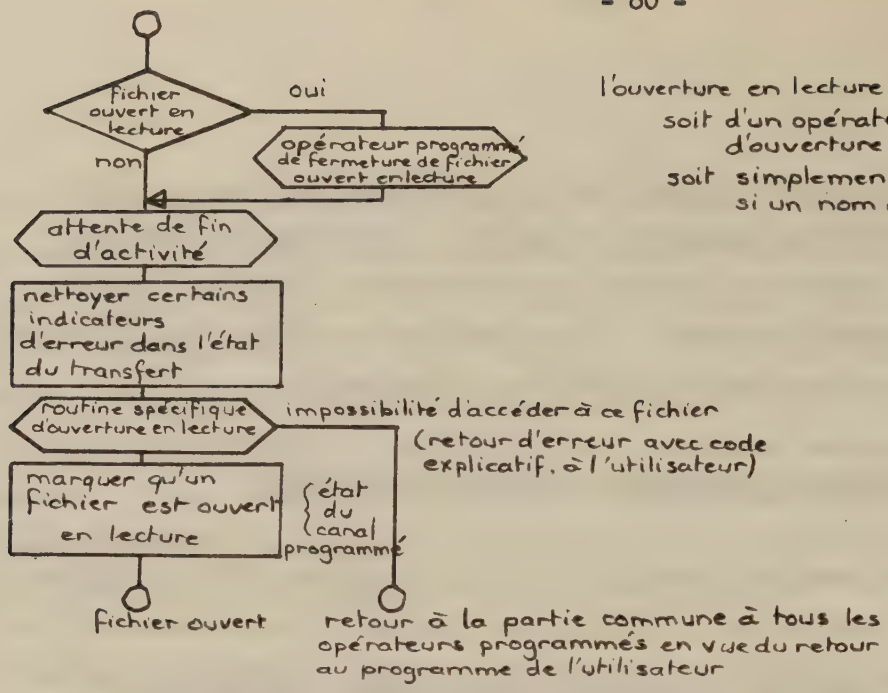
La non précision d'un numéro d'enregistrement entraîne automatiquement un transfert séquentiel. Un opérateur permet de préciser le numéro du prochain enregistrement à lire ou à écrire sur l'équipement concerné, il ne s'applique en fait qu'aux fichiers sur disques ou aux bandes magnétiques adressables.

7 - Opérateurs programmés d'accès à un fichier

Le nom du fichier, le nom du répertoire, s'il n'est pas implicite, dans lequel le fichier va être lu ou écrit doivent être précisés en arguments des opérateurs d'accès (ouverture en lecture ou en écriture). Seul le nom est indispensable, sur le disque, l'absence de nom de répertoire signifie la propre zone de l'utilisateur ; d'autres quantités sont retournées automatiquement : date de création, de dernière consultation, heure et mode de création, privilèges d'accès.

Comme l'initialisation d'un équipement provoque la libération de l'équipement préalablement initialisé sous le même numéro de canal programmé, l'ouverture d'un fichier en lecture ferme le fichier déjà ouvert au préalable en lecture si l'ordre de fermeture a été omis. En écriture, le fichier est également fermé mais il est en général plus prudent de contrôler minutieusement que les opérations élémentaires se sont bien passées (absence d'erreur d'écriture par exemple).

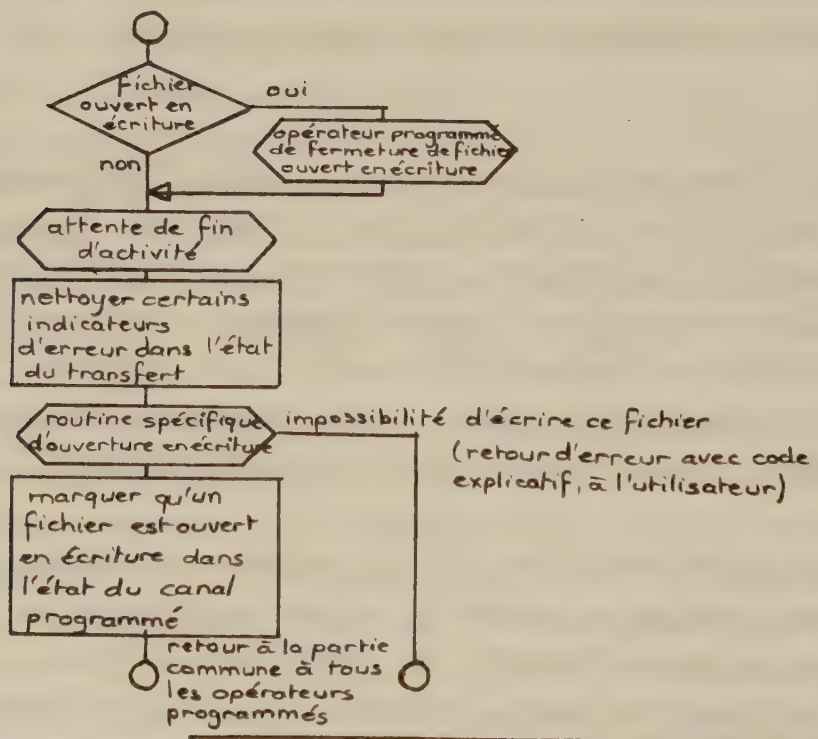
L'opérateur n'a réussi que s'il y a au retour du décodeur, saut d'une adresse; dans le cas contraire, un code est retourné au programme motivant le refus d'ouverture du fichier (non existence, accès protégé, modification en cours, impossibilité physique d'écrire dans un répertoire borné saturé). Le programme peut prendre les décisions qui s'imposent dans les différents cas possibles sans être pour autant interrompu.



l'ouverture en lecture résulte
soit d'un opérateur programmé d'ouverture précédent
soit simplement d'un ordre d'entrée si un nom de fichier n'est pas requi

Routine générale de décodage d'une ouverture de fichier en lecture

---



Routine générale de décodage d'une ouverture de fichier en écriture

8 - Opérateur programmé d'entrée

A l'initialisation de l'équipement, un certain mode de transfert a été précisé, ce mode dépend parfois de l'équipement logique connecté par cette opération mais on peut distinguer :

I°/ - les modes avec tampons habituels soit mot par mot, soit caractère par caractère avec ou sans arrêt en fin de ligne, soit sans conversion en code caractère pour des supports spéciaux tels que carte ou ruban papier avec contrôles de parité,

2°/ - les modes sans tampon (directs), pour lesquels le programme doit fournir un mot de commande donnant l'emplacement et le nombre de mots à transférer.

Le mode de transfert peut très bien être changé en cours de travail grâce à l'opérateur de modification d'état du transfert en principe destiné à effacer les indicateurs d'erreurs. Le mode direct est réservé aux équipements pour lesquels le superviseur exécute lui-même un transfert mot par mot, par mot de commande et non par assemblage ou désassemblage des mots en caractères : disques, bandes magnétiques mais pas le lecteur de cartes ou les consoles des utilisateurs. Le mode direct nécessite pour le programmeur une gestion complète des enregistrements qu'il traite, la connaissance de leur place et de leur longueur, ceci est très utile pour les applications particulières telles que l'écriture ou la lecture sous des formats inhabituels comme des enregistrements de bandes magnétiques compatibles avec d'autres installations ou plusieurs blocs de disque d'un seul coup.

a) - remplissage asynchrone des tampons

L'utilisation des tampons comme zones implicites d'entrée et de sortie permet de soulager l'utilisateur du contrôle de ses enregistrements, surtout s'il désire effectuer une lecture séquentielle de ses données rangées dans un fichier. Ceci permet aussi d'imbriquer les opérations de transferts effectuées par le superviseur en régime d'interruption prioritaire avec le déroulement du programme concerné par ces opérations de transfert, ce dernier programme effectuant des traitements sur ses données déjà fournies. L'hypothèse du traitement séquentiel permet au superviseur, si aucune contrainte ne s'y oppose, de prendre de l'avance sur le programme en lui transférant dans des tampons libres les données dont il aura besoin dans le futur.

Dans ces conditions, l'opérateur programmé ne fera qu'une simple manipulation des pointeurs de l'utilisateur sur la suite de son fichier déjà lu dans le tampon qui suit celui qu'il vient de traiter et qui ne l'intéresse plus.

L'utilisation des tampons conduit à l'asynchronisme du traitement par le programme et du transfert par le superviseur en régime d'interruption, le plus rapide des deux doit attendre le plus lent : arrêt de traitement = arrêt de déroulement si le transfert est trop lent, arrêt de transfert si tous les tampons sont pleins dans la boucle parce que le traitement est trop lent.

Le programme, pour utiliser au mieux ces tampons, doit pour accéder au caractère suivant le plus facilement utiliser la séquence :

. décrémenter le compte de caractères restant et sauter si positif,

. appeler le tampon suivant et vérifier l'absence d'erreur,

. charger le caractère suivant, dans un registre, en incrémentant le pointeur de caractère fourni par le superviseur

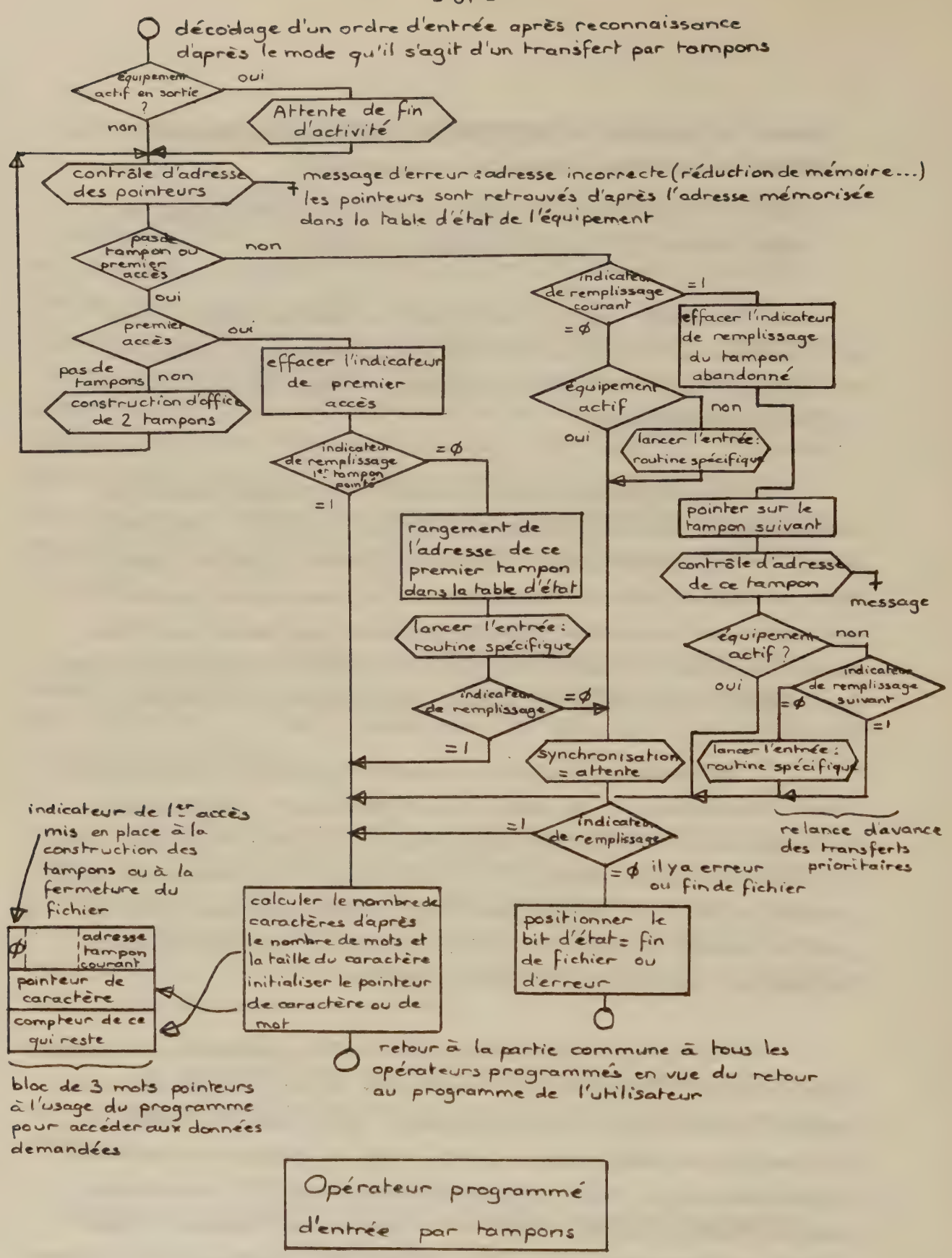
soit avec 2 instructions + saut au sous-programme d'appel du tampon suivant qui teste l'absence d'erreurs.

Le compte étant nul au départ, le premier appel se fera de lui-même, il est inutile de prévoir ailleurs dans le programme des ordres d'entrée de tampon préalables, l'accès au caractère (mot) le fera automatiquement.

Lors du premier appel de tampon, les tampons peuvent ne pas exister, 2 sont alloués automatiquement à la fin du programme. Sinon lors du premier accès l'adresse du premier tampon est rangé dans la table d'état de l'équipement et le processus de transfert est amorcé par la routine d'entrée de données par tampon spécifique de cet équipement, à laquelle on accède au travers de la table d'état qui contient l'adresse de sa table de branchement par fonctions.

Un transfert d'entrée se termine, en général, en interruption prioritaire, par la mise à 1 de l'indicateur de remplissage du tampon sauf si le fichier est terminé ou que des erreurs invalident le contenu du tampon actuel. Si le remplissage s'est bien passé, après attente de la fin du premier transfert, les pointeurs sont remis à jour en fonction du nombre de mots transférés et pas toujours de la longueur du tampon qui peut y être supérieure.

Pour qu'il n'y ait jamais d'ambiguïtés, précaution est prise avant tout test d'activité de l'équipement, qu'il n'y a pas activité pour un transfert dans le sens contraire, ici le même équipement ne doit pas être en même temps en sortie quand on risque de demander de nouveau une entrée. Cette précaution étant prise, s'il y a activité dans les tests suivants, c'est que des transferts d'entrée se poursuivent au niveau des interruptions prioritaires emplissant des tampons en avance sur l'utilisateur.

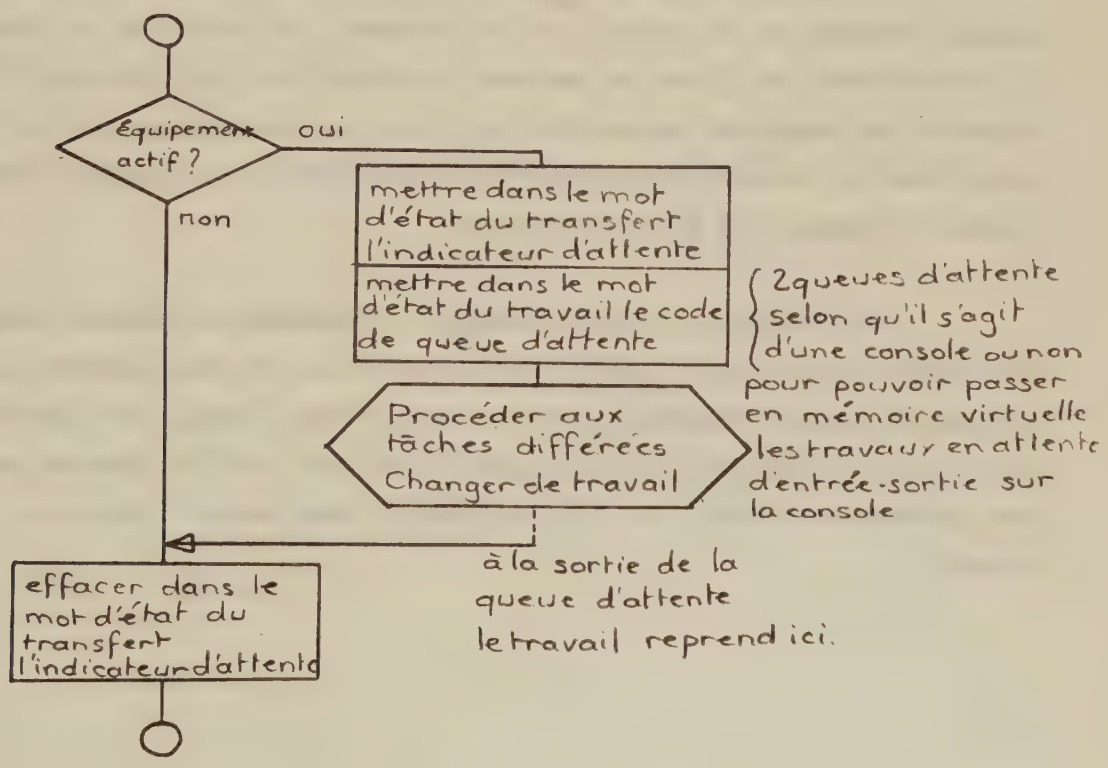


Opérateur programmé d'entrée par tampons

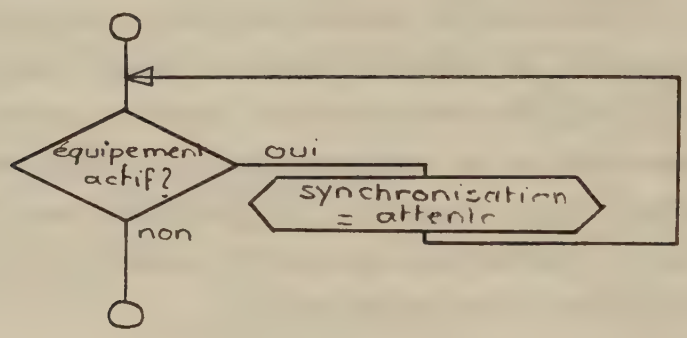
Dans ces conditions, il est inutile de relancer une entrée par la routine spécifique de l'équipement car au moins un tampon est prêt ou sera prêt à la première fin de transfert. Dans ce dernier cas il faut faire appel au mécanisme de synchronisation qui conduit à l'attente de la première fin de transfert dans une queue d'attente de fin d'entrée-sortie avec changement forcé de travail courant, on en sortira sur signalisation de changement de queue vers queues de passage à la première fin de transfert, le changement ayant lieu tout de suite si c'est le travail nul qui a pris la main ou à la première interruption d'horloge dans le cas contraire.

b) - routines prioritaires

Au niveau de l'interruption prioritaire, à la fin du transfert, la routine est prévenue que le travail qui a lancé le transfert désire sortir de sa file d'attente par une indication logée dans le mot d'état du transfert par la routine de synchronisation.



Le module d'attente de fin d'activité fait essentiellement un appel répété au module de synchronisation :



Cette façon de procéder est indispensable pour ne pas manquer de détecter la première cessation d'activité.

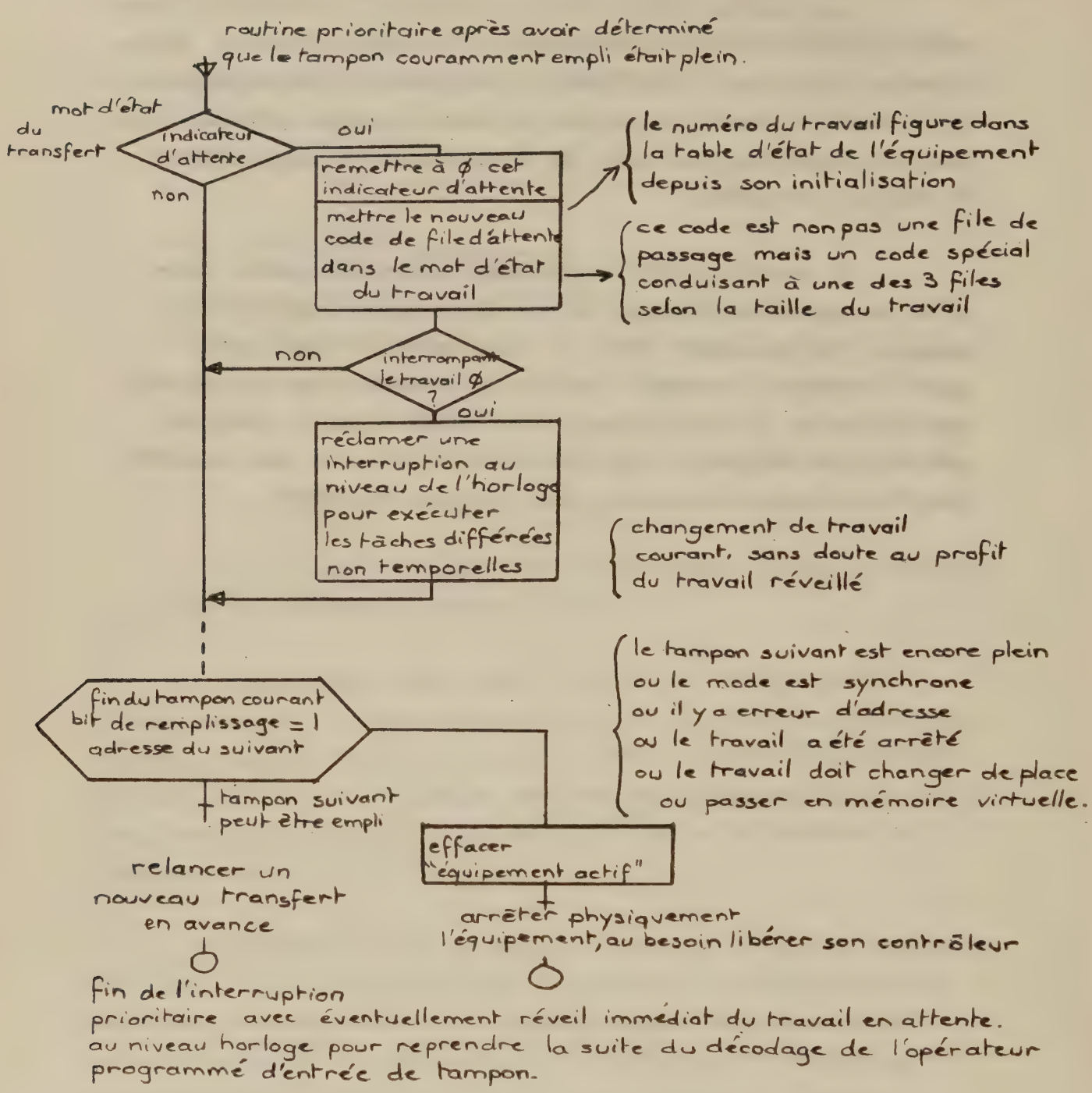
Les routines spécifiques, à la fin du remplissage d'un tampon doivent, si le travail est en attente, le réveiller et provoquer le retour dans les files de passage. Le réveil est une tâche différée signalée au programme de gestion des tours de passage par un bit du mot d'état du travail positionné une seule fois et un compteur des travaux à changer de file en différé.

Le retour en file de passage se traduira au premier changement de travail fait à notre profit par la poursuite du déroulement de la routine de synchronisation, avec ou sans l'état "équipement actif" mais avec au moins le tampon suivant transféré, ici en mémoire ou une indication d'erreur ou fin de fichier dans le mot d'état du transfert.

Les routines prioritaires qui traitent les demandes d'interruption doivent lorsque le remplissage d'un tampon est achevé, satisfaire aux conditions d'enchaînement de transfert au tampon suivant et de réveil du travail en attente au moyen de modules généraux réalisant ces algorithmes pour tous les périphériques de façon uniforme et bien sûr réentrante.

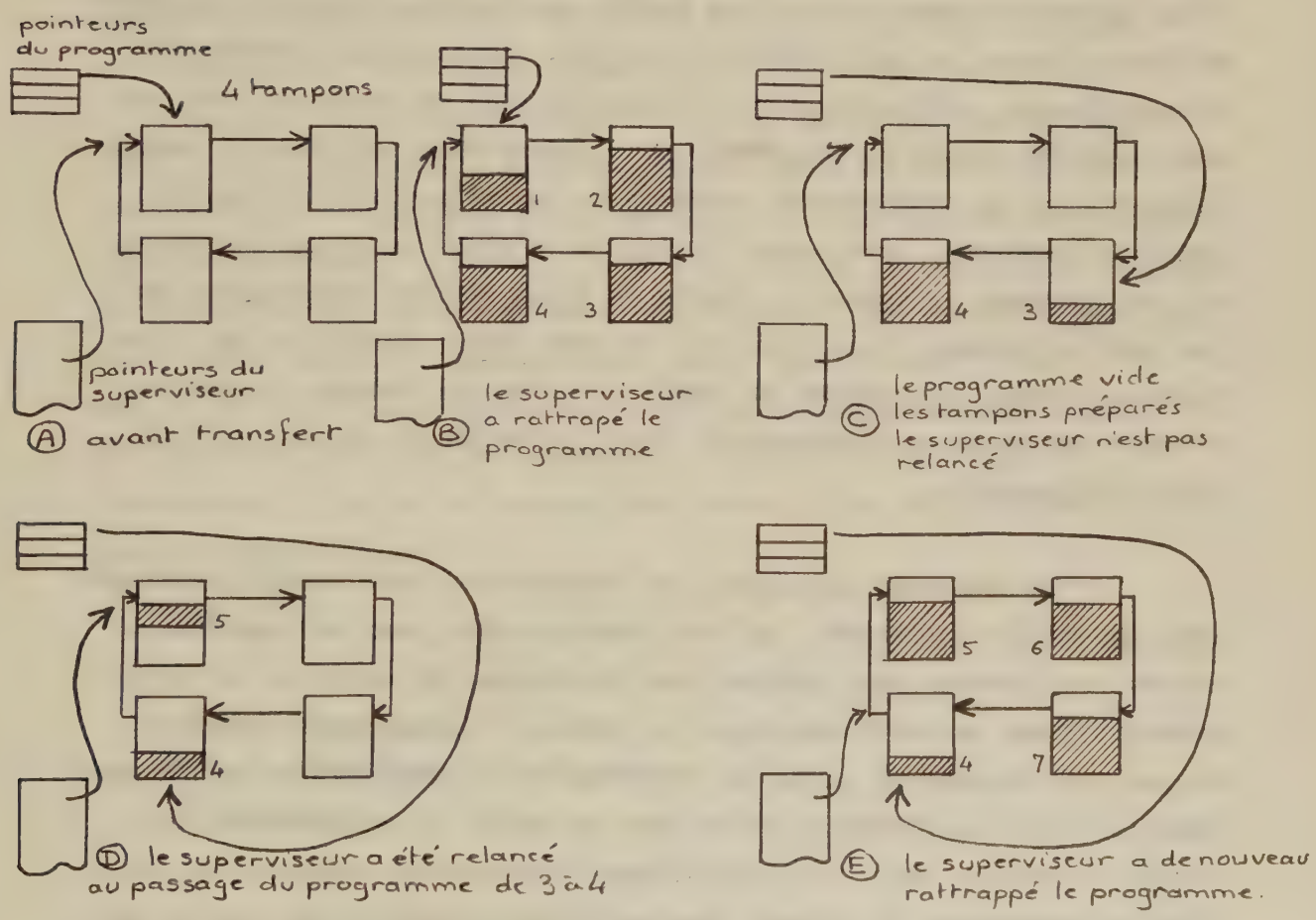
Un certain nombre de conditions favorables peuvent faire qu'un équipement actif puisse se poursuivre après transfert d'un tampon et éventuellement réveil et relancement du travail qui a lancé l'opérateur programmé d'entrée. Les tampons vides qui suivent, dans la boucle, le tampon venant d'être empli et que le programme peut utiliser désormais, vont être emplis à leur tour sans exception jusqu'à rattraper le programme dans la boucle des tampons.

Achèvement d'un transfert en interruption prioritaire



c) - problèmes de synchronisation - remplissage - utilisation

La rupture de l'avance prise par le superviseur se détecte lorsqu'en avançant d'un tampon, les pointeurs du programme lançant l'opérateur programmé, on constate que ce tampon n'est pas déjà rempli, le processus de remplissage est alors relancé par la routine spécifique et il faudra sans doute de nouveau attendre. Dans le cas où le tampon avancé est déjà plein, un contrôle simple permet de voir si le suivant est également déjà plein et s'il ne l'est pas, et sauf s'il s'agit de l'équipement "console" qui ne doit pas prendre d'avance sur le programme, le processus de remplissage est relancé jusqu'à, par exemple rattraper le programme.



d) - utilisation des contrôleurs de périphériques par "bouffées"

Comme le montrent les schémas A, B, C, D, E, les transferts ont lieu non pas continûment mais par bouffées si le débit du périphérique est supérieur à la vitesse de traitement du programme, ces bouffées sont d'ailleurs d'autant plus longues qu'il y a plus de tampons et que les vitesses sont voisines.

Cette caractéristique est très avantageuse pour l'usage des périphériques dont la mise en action est longue relativement aux transferts proprement dits : accès séquentiels sur bande ou disque magnétique ou dont la connexion à un contrôleur est liée à la liberté de ce contrôleur s'il est partageable. C'est avec cette méthode, si la boucle porte au moins 2 tampons, que les contrôleurs de périphériques multiples sont les mieux utilisés car tout transfert en train, s'il peut être poursuivi, l'est systématiquement et ne sera repris que pour un transfert également de longue durée, ce qui réduit au total les changements d'attribution de contrôleurs, coûteux surtout en temps de mise en route de l'équipement contrôlé (bande en mouvement, bras disque bien positionné). Dans certains cas, un accaparement excessif du contrôleur est à craindre avec cette méthode, prolongeant l'attente de cette ressource partageable et diminuant le temps de réponse de façon surtout appréciable par les petits travaux, c'est pourquoi, on peut alors limiter en cas d'attente le temps d'occupation ou plutôt le volume de données transmises d'avance au programme (automatique dans la routine de gestion de certains équipements).

Ce mécanisme asynchrone de transferts et traitement simultanés dans une boucle de tampons, qui est indispensable dans les systèmes à un seul utilisateur est conservé pour le partage de temps en lui adjoignant quelques restrictions visant à libérer l'accaparement possible de quelques ressources comme les contrôleurs de périphérique ou l'espace mémoire quand les entrées-sorties sont en cours. La simultanéité complique la gestion mais est très efficace puisqu'elle consiste ici à profiter au maximum de la liberté des contrôleurs d'abord, de l'unité centrale ensuite dès qu'ils sont disponibles. Tous les algorithmes s'appliquent au cas d'un seul tampon pointant sur lui-même.

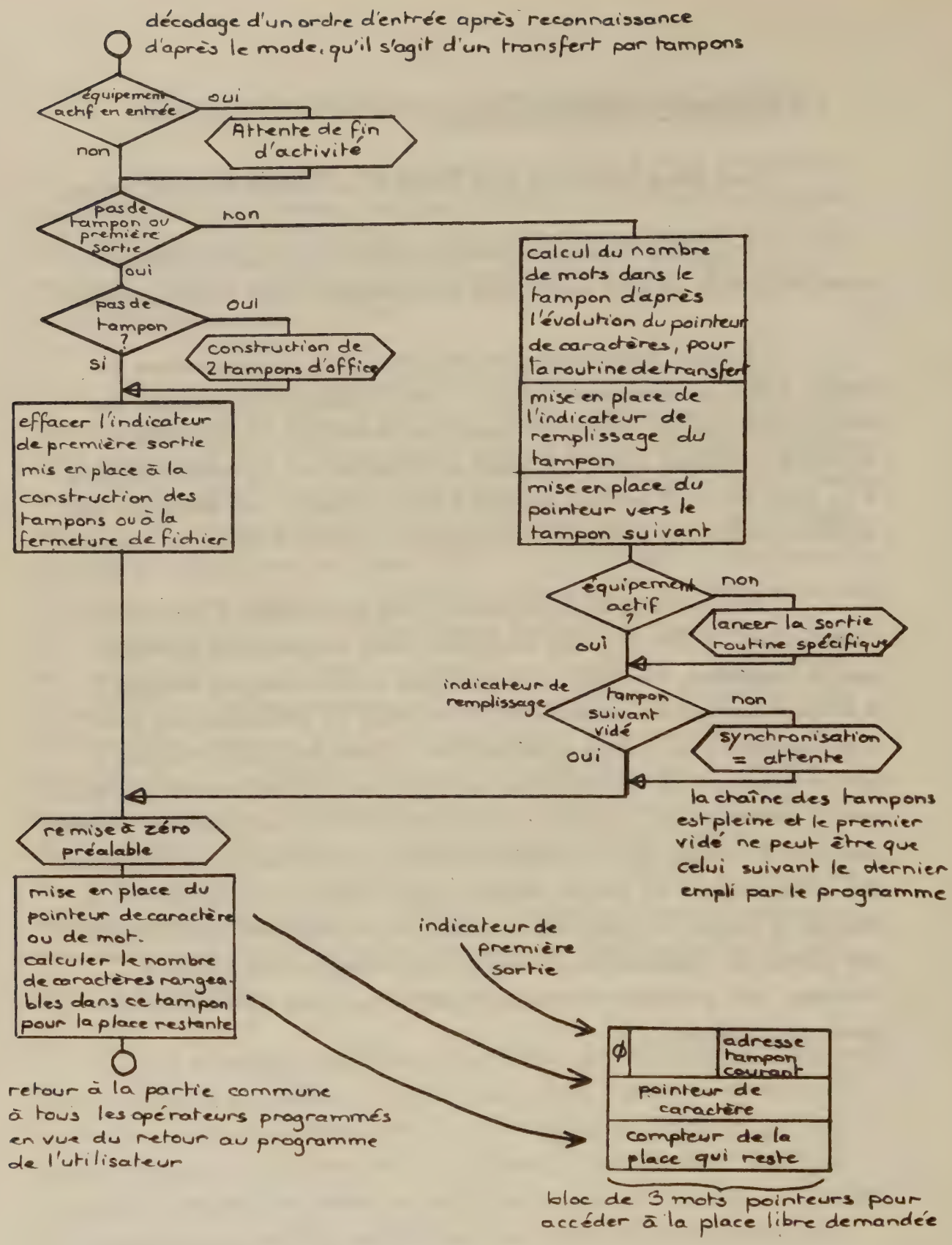
9 - Opérateur programmé de sortie

Les mêmes principes sont utilisés en sortie comme en entrée.

Les modes directs existent également, ils sont synchrones comme en entrée et leur usage plus contraignants pour la programmation.

En mode interruption prioritaire, lors du constat que le tampon a été convenablement vidé, il est nécessaire de procéder aux mêmes opérations de réveil du travail en attente de tampon vide et d'avance au tampon suivant à vider si l'indicateur de remplissage est à 1. Ceci est fait par une routine à deux retours, pour continuer ou arrêter, similaire à celle qui avance les tampons d'entrée en séquence.

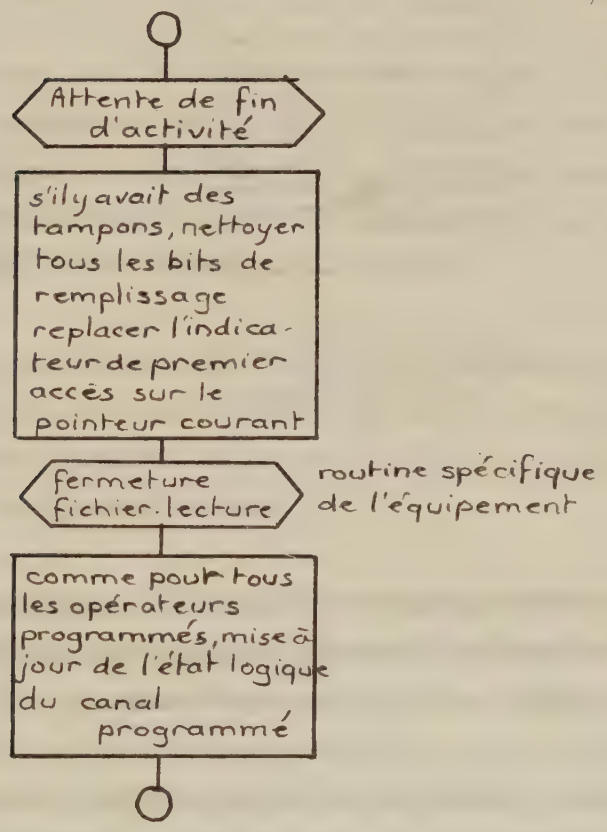
Tant en entrée qu'en sortie, les algorithmes d'avance de tampons sont conçus de telle sorte que toute manipulation anormale par le programme, des bits de remplissage situés dans les tampons à l'usage surtout du superviseur, n'entraîne de préjudices que pour l'utilisateur sans jamais entraîner un blocage possible comme une synchronisation improbable et un maintien prolongé en attente d'entrée-sortie immobilisant le travail en mémoire réelle. Les contrôles d'adresses sont effectués par le décodeur d'opérateurs programmés et au niveau des enchaînements au tampon suivant, mais, dans ce cas, la sortie de message d'erreur est reportée au moment où le transfert sera relancé par opérateur programmé au niveau non prioritaire. (Les routines d'erreur sont à éviter en niveau prioritaire, elles sont autant que possible différées).



Opérateur programmé de sortie par tampons

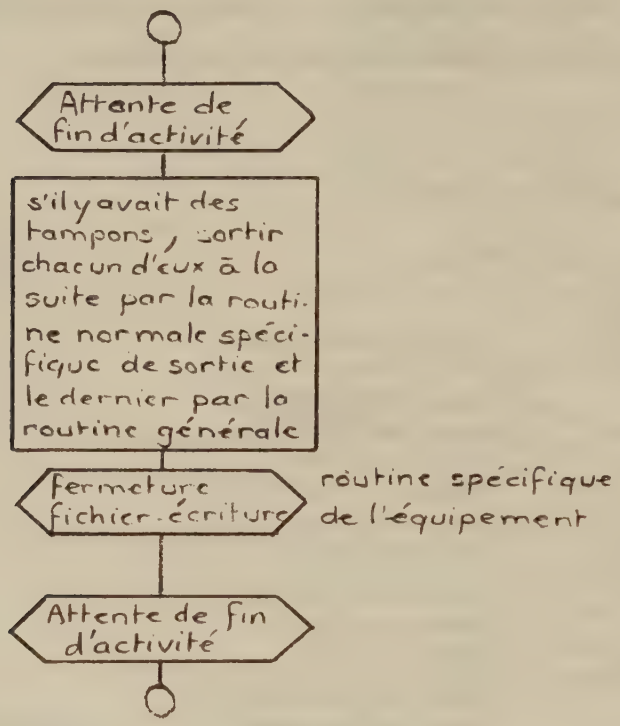
10 - Opérateur programmé de fermeture de fichier en entrée

Il existe un seul opérateur pour fermer l'entrée et la sortie sur un même équipement avec ou sans, d'ailleurs, ouverture préalable de fichier puisque des transfert peuvent exister avec d'autres supports logiques que des fichiers (cartes, imprimante). Ce même opérateur peut inhiber une des fermetures, soit l'entrée du fichier d'entrée, soit du fichier de sortie (soit les 2, auquel cas c'est une non-opération). L'usage le plus simple par omission des inhibitions ferme l'entrée et la sortie.



En fait, il n'y a pas passage dans la routine spécifique si aucun transfert n'a été entrepris (simple ouverture de fichier sans consultation pour vérifier l'existence ou la date de ce fichier) sauf s'il s'agit des disques pour lesquels à toute consultation la dernière date de référence est répertoriée.

11 - Opérateur programmé de fermeture du fichier en sortie



Le dernier opérateur programmé de sortie peut être omis, tous les transferts non terminés sont achevés et le dernier tampon, s'il n'est pas vide, est sorti automatiquement.

En cas d'écriture de fin de fichier, de nouveaux transferts ont lieu, pour la suite, il faut que toute activité ait cessé.

12 - Opérateur programmé de libération d'équipement

Cet opérateur procède avant l'abandon de l'équipement à la fermeture (éventuelle) des fichiers ouverts en lecture et en écriture sur cet équipement (forçant l'écriture du dernier tampon en cours de remplissage par le programme). Selon les équipements la libération remet à jour la table d'état de cet équipement dans les parties qui en sont spécifiques et éventuellement libère le contrôleur en usage et demande que le premier travail en attente de ce contrôleur soit réveillé.

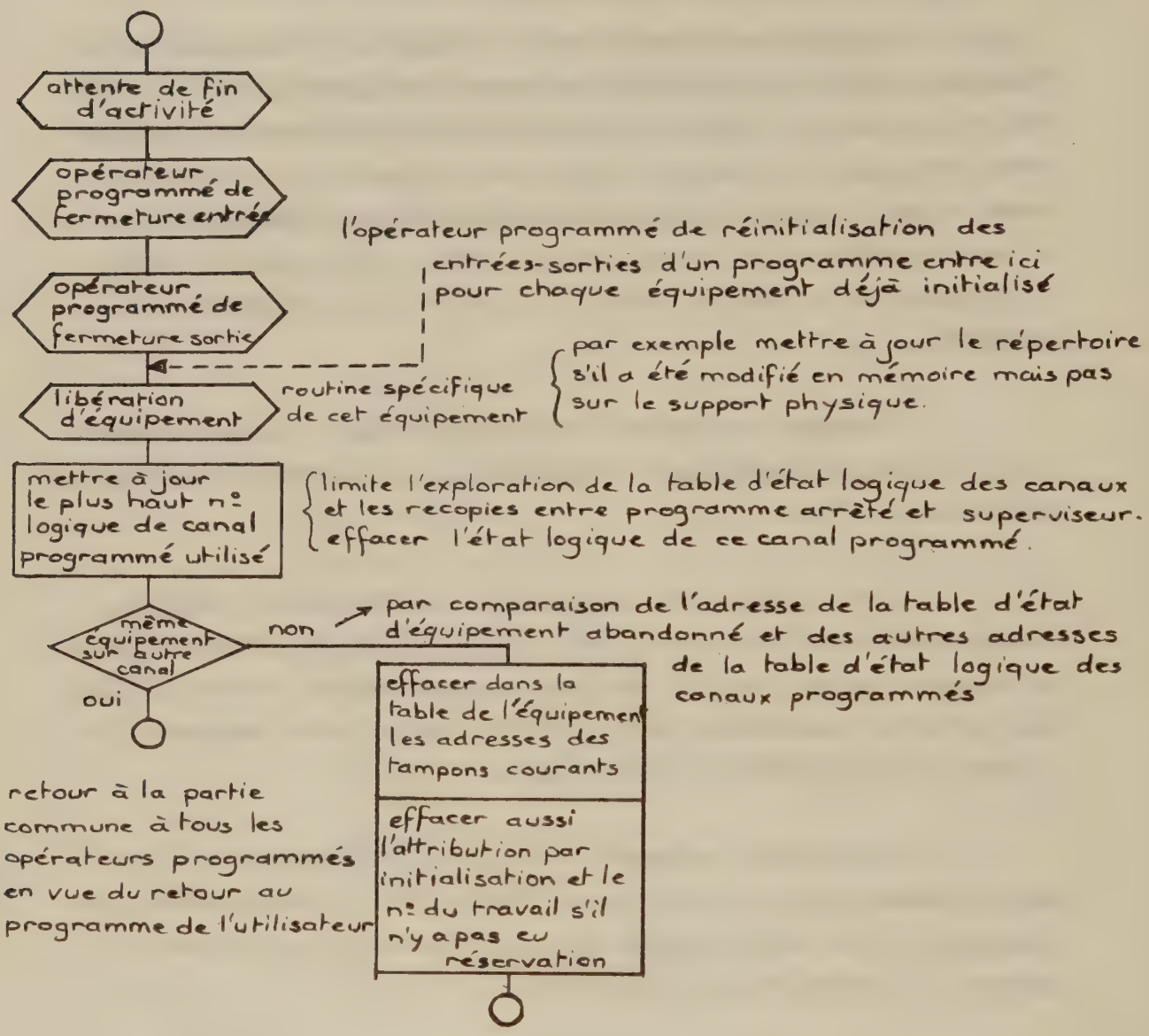
L'adresse des tampons courants est mise à jour avant abandon s'il n'a pas été réservé par l'utilisateur, de l'équipement représenté par sa table d'état.

Un équipement non initialisé et non réservé appartient au domaine public des ressources partageables. Il est en général conseillé de réserver les équipements sur lesquels des fichiers personnels sont lus ou écrits pour éviter leur altération par un autre programme à la libération programmée de cet équipement.

Un opérateur programmé de retour au superviseur avec arrêt du programme, assure la fonction de libérer tous les équipements initialisés. Il est ainsi possible de ne pas se donner la peine de programmer la libération des équipements.

L'opérateur programmé de réinitialisation des entrées-sorties signalé à propos des zones de tampons assure la libération des équipements initialisés sans procéder à la fermeture des fichiers ouverts considérés comme sans valeur, en particulier en sortie. Certaines routines de libération spécifiques devront se prémunir contre ce cas d'appel (lorsque des blocs de disques avaient été réservés pour écrire un nouveau fichier, il faut faire disparaître toute trace de cette tentative). C'est ce dernier opérateur qui est également utilisé par

Opérateur programmé d'abandon d'équipement



le superviseur lorsqu'un travail est arrêté par l'ordre d'arrêt issu de la console de l'utilisateur ou par un diagnostic du superviseur, et suivi d'un ordre appelant un nouveau programme sans laisser se terminer le programme précédent.

Avant de lancer ce nouveau programme, le précédent doit avoir libéré les équipements déjà initialisés, cette condition est l'une de celles qui vérifie et que fait remplir le programme décodeur des ordres préalablement à l'exécution proprement dite et éventuellement après attente de transferts rendus nécessaires par la libération de l'équipement.

Toutes les routines exposées jusqu'ici assuraient des fonctions strictement logiques à peu près indépendantes des équipements physiquement concernés pour lesquels les fonctions spécifiques étaient exécutées par appel sur pile à une adresse retrouvée grâce à la table d'état.

Certains équipements supports de fichiers jouissent d'un avantage supplémentaire sur les autres équipements : la possibilité d'accès simultané au même support et au même fichier par des travaux différents tournant en concurrence dans le système. L'avantage de cette possibilité est d'augmenter la virtualité de la machine puisque en plus du matériel et du temps de calcul, des fichiers, c'est-à-dire des données, sont quasi-simultanément accessibles et entrent eux aussi dans les ressources partageables. Les équipements qui entrent dans cette catégorie sont adressés sous le générique DISQUE qui peut recouvrir des organes aussi différents que des disques à tête fixe ou des tambours (servant par exemple également de support à la mémoire virtuelle) ou des piles de disques éventuellement amovibles.

Parmi les ressources partageables d'un système, les possibilités les plus récentes et les plus évolutives se trouvent dans les structures de fichiers à accès multiple par plusieurs travaux simultanés d'un système en partage de temps. C'est pourquoi les routines spécifiques de l'organe général DISQUE méritent une étude particulière.

X - UTILISATION D'UN EQUIPEMENT PARTAGEABLE MULTIPLEMENT ACCESSIBLE

L'étude va être limitée à celui d'un seul disque public, les procédures étant appelées à s'étendre au cas de plusieurs organes du type DISQUE publics se faisant suite du point de vue logique au fur et à mesure du remplissage et de DISQUES privés accessibles par un seul travail : celui qui l'a réservé (cas d'une pile amovible (et de son contrôleur) réservée, montée, mise à jour, démontée, libérée).

Les objectifs initiaux d'une structure de fichiers peuvent être résumés à :

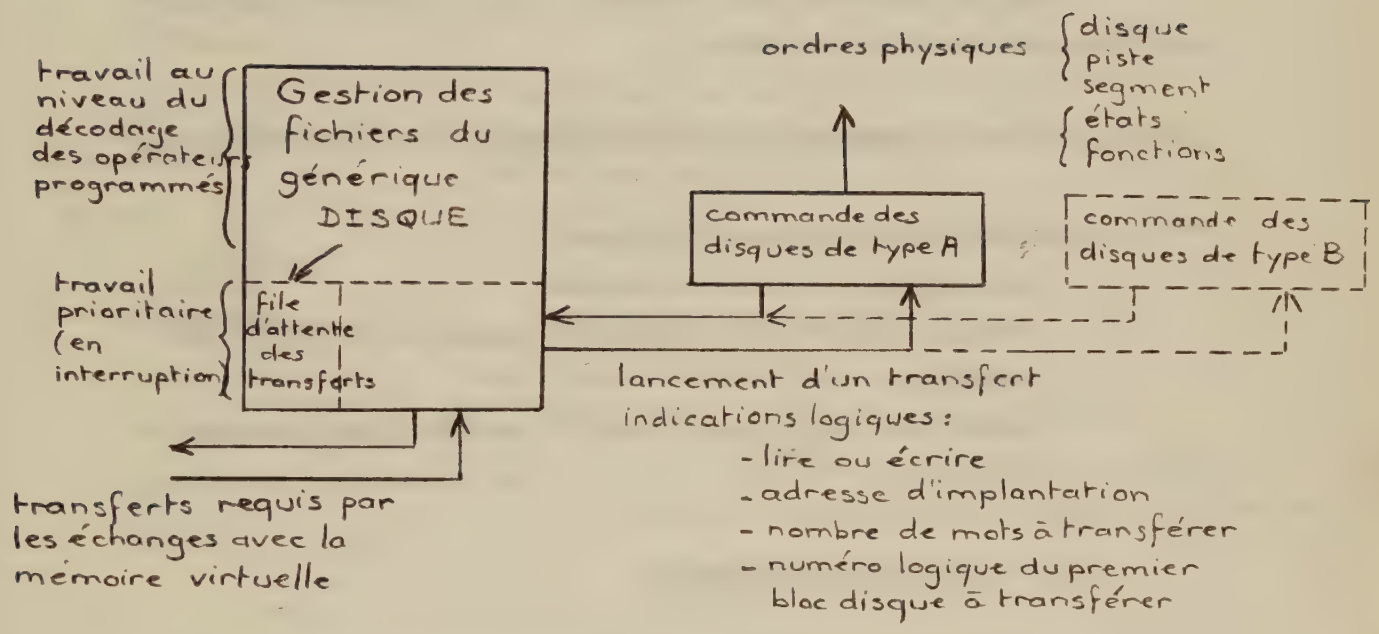
1°/ - non limitation tant sur le nombre que sur le volume des fichiers autre que la capacité physique de stockage d'information,

2°/ - facilité d'accès à l'information,

3°/ - contrôle suffisant de la validité de l'information relue sans accorder une confiance excessive au matériel.

L'optimisation du temps de recherche et de transfert de blocs consécutifs n'a pas été du tout un objectif essentiel ; on cherche certes, à allouer des blocs libres consécutifs à l'écriture d'un même fichier mais en général l'accès est fait par blocs individuels. Une simple file d'attente constituée au niveau du décodage des opérateurs programmés pour les ordres de transferts déjà préparés est utilisée au niveau prioritaire d'interruption pour fin de transfert sans prévoir le nombre de rotations ou de mouvements de bras du disque avant d'accéder à l'enregistrement désiré suivant dans la file d'attente. Le module de gestion du générique DISQUE fait abstraction de la réalité physique du contrôleur de disque utilisé, c'est une routine spécifique de l'équipement physique qui est chargée de lancer et contrôler le transfert avec (ou sans) un canal d'accès direct à la mémoire.

1 - Organisation logique



La routine spécifique de commande de l'équipement physique (contrôleur) assure les changements de coordonnées (numéro logique de bloc en disque, (face), piste, segment) et les codages des ordres de transfert conformément aux spécifications physiques de l'équipement. La possibilité d'étendre le contrôle à plusieurs types d'équipements suppose simplement que les blocs recouvrent ou peuvent recouvrir le même multiple de mots pour qu'un numéro logique ait un sens sans perdre

de place sur le plus petit élément adressable du support extérieur. En pratique, un bloc contient 128 mots, c'est la plus petite unité adressable sur un équipement (il s'agit par exemple de 4 segments de 32 mots de 36 bits, chaque segment portant son contrôle de parité longitudinale).

2 - Nature et hiérarchie des informations stockées

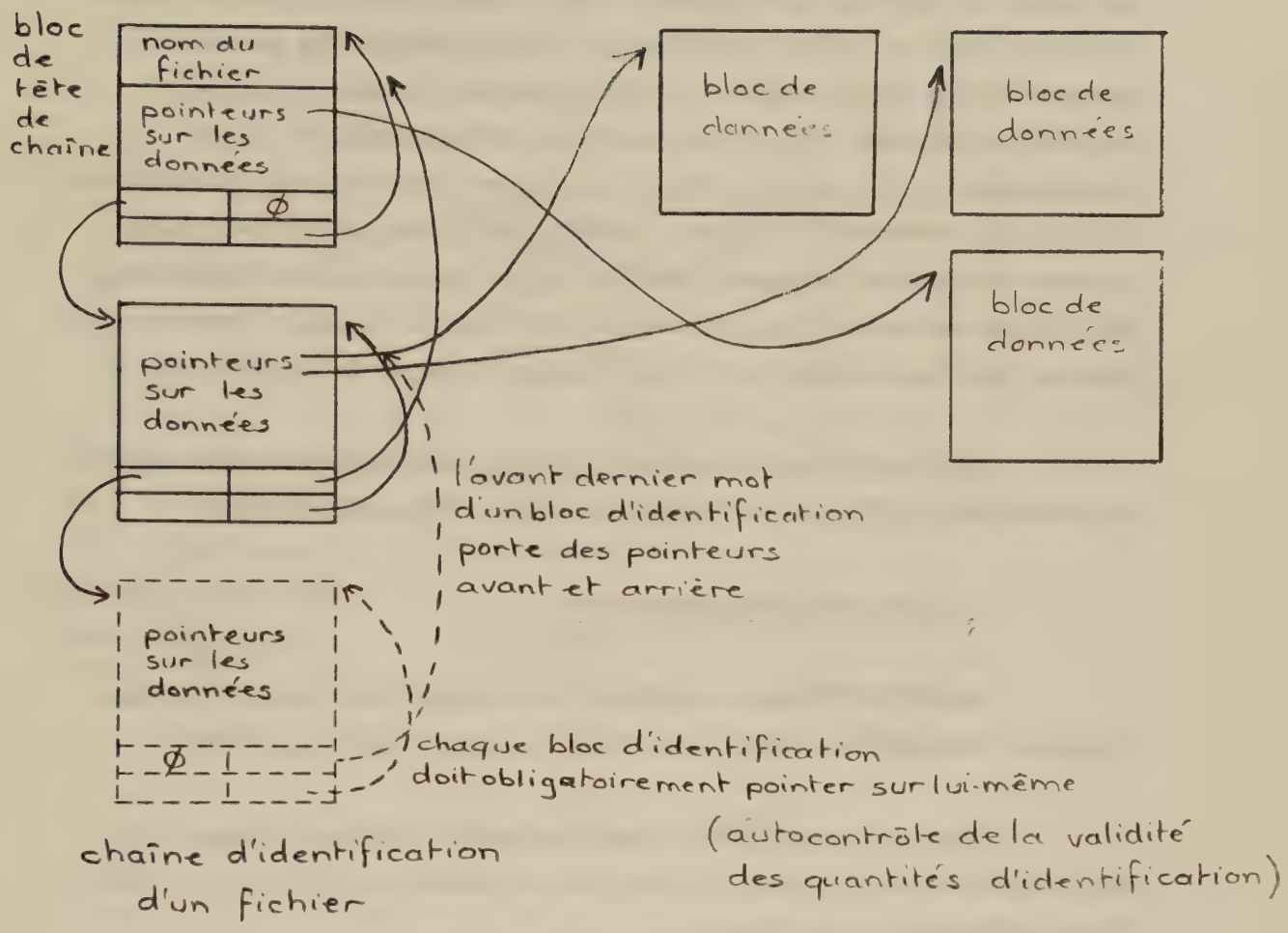
Dans la structure des fichiers, il y a 2 genres d'éléments bien distincts quant à leur rôle :

1°/ - les blocs de données n'ont pas d'information d'identification,

2°/ - les blocs d'identification indiquent :
- quels sont les blocs de données,
- quelles conditions de validité chacun de ces blocs doit vérifier,
- quels sont les autres blocs d'identification associés à celui-ci.

a) - bloc d'identification

Un seul mot suffit à identifier un bloc de données, ce mot contient : la somme cyclique du contenu du bloc au moment de l'écriture (un demi-mot),
le numéro logique du bloc en question (un demi-mot permet d'indiquer 256. K blocs de 128. mots).



Les blocs logiques ont 128. mots, ils contiennent 126. (122. pour le premier) pointeurs sur des blocs ayant également 128. mots de données, soit environ un volume de 16. K mots identifié par un seul bloc d'identification. La structure adoptée n'est pas limitative quant au volume d'un fichier grâce au chaînage des blocs d'identification.

Les blocs de données sont ordonnés par la position relative de leurs pointeurs dans la chaîne des blocs d'identification. Le premier bloc souvent unique de la chaîne d'identification n'a pas de pointeur arrière et contient à la place des 4 premiers pointeurs le nom (9 caractères = 1,5 mot), la date d'accès, un code d'accès (protection), la date et l'heure de création, le mode utilisé à l'écriture et la longueur du fichier en mots. Sauf la protection et le nom, ces caractéristiques ne servent pas au superviseur mais à l'usager accédant au fichier dont la longueur est ainsi connue d'emblée à l'ouverture.

Un fichier par lui-même est accessible dès que l'on connaît le numéro de la tête de la chaîne des blocs d'identification.

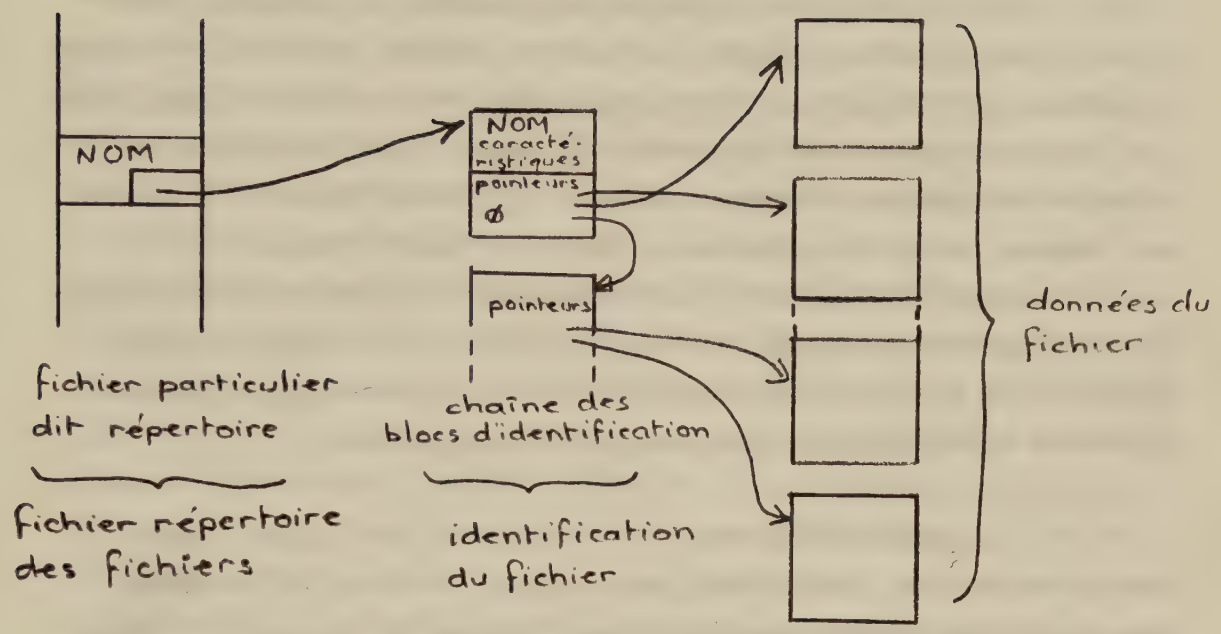
b - les fichiers répertoires

Certains fichiers reçoivent un contenu particulier faisant d'eux des répertoires tenus à jour par le superviseur lui-même.

Dans un répertoire, 2 mots servent à accéder à chaque fichier répertorié : le nom (1,5 mot) et le numéro de tête de la chaîne d'identification de ce fichier.

64 fichiers peuvent être rendus accessibles par un seul bloc de fichier répertoire qui possède lui aussi un bloc d'identification (ou une chaîne pour plus de 64 x 122. fichiers répertoriés !).

En pratique, chaque utilisateur inscrit dans le système, possède un fichier répertoire de longueur quelconque pointant sur tous ses fichiers. Un utilisateur peut manipuler ses fichiers à son gré mais pas son fichier répertoire dont la tenue ressort du système lui-même.



Ce fichier répertoire appartient au système, un utilisateur peut lire ce fichier mais non y écrire (code d'accès).

c) - fichier répertoire général

Pour accéder aux fichiers répertoires, au moyen de leurs blocs d'identification propres, le système possède lui-même un fichier répertoire général unique répertoriant pour une structure disque, tous les fichiers répertoires (y compris lui-même !) qui comme tous ses fichiers ordinaires, lui appartiennent. Ce fichier répertoire général est reconnu par un bloc (ou une chaîne !) d'identification dont le numéro logique de tête doit nécessairement être connu du superviseur. Ce renseignement est rangé dans un bloc implicite de chaque disque (bloc vital) et lu avec la capacité et le type de ce disque, automatiquement à la première initialisation du superviseur.

La demande de lecture ou d'écriture d'un fichier (nom) dans un répertoire (nom de son propriétaire) se fera par recherche dans le fichier répertoire général de l'adresse de la chaîne d'identification du fichier répertoire, consultation de ce répertoire à la recherche du fichier et lecture du premier bloc d'identification pour vérifier les modalités d'accès et permettre la lecture du premier ou préparer une première écriture.

Pour créer un fichier, la recherche des blocs non alloués sur le support se fait par consultation d'un fichier réservé au système et qui constitue la table d'occupation du disque en question. Ce fichier particulier, que seul le système modifie, est répertorié dans le répertoire général. Au début de l'histoire du disque (procédure de mise en place d'un disque hors exploitation), le fichier répertoire général et son bloc d'identification, le fichier d'occupation du disque marqué des blocs déjà réservés, des blocs non attribuables car défectueux et le bloc vital sont écrits sur le disque. L'essentiel de la structure existe désormais sur le disque pour sa croissance en exploitation.

L'opérateur chargé de lancer un superviseur avec un disque ainsi démuni lorsque le disque doit aussi servir de bibliothèque système dispose d'un privilège l'autorisant à travailler sans s'inscrire sur le livre de bord, sans passer par la routine normale de mise à jour du livre de bord par un programme implicitement appelé par le superviseur dans la bibliothèque système. Le livre de bord est un fichier appartenant au système et rangé, protégé contre les modifications par les étrangers sur le disque. Le premier travail est de charger la bibliothèque système au moyen de programmes de service normaux et d'ouvrir un livre de bord au moyen des programmes spécifiques de sa gestion. Les fichiers répertoires sauf le - général, n'existent pas et seul le système ou un programme travaillant en son nom peut les ouvrir (les créer) dans le fichier répertoire général. Cette tâche est faite lors de la procédure d'entrée dans le système d'un utilisateur, lorsqu'on met à jour le livre de bord et que temporairement, l'utilisateur ne dispose que des commandes minimales sous contrôle d'un programme implicite qui travaille au nom du système.

S'il n'existe déjà, un nouveau fichier, futur répertoire pour l'utilisateur, est créé, qui porte son nom et qui répertoriera tous les fichiers de ce client.

La spécialisation de certains fichiers en répertoires a permis sans restriction de dimension et de nombre de rendre facile l'accès à tout fichier au moyen de 2 recherches en cascade dans des répertoires par localisation d'abord, du propriétaire, puis du fichier ; la consultation d'un répertoire est effectuée par une routine unique du superviseur. La lecture d'un bloc de données (fichier simple ou fichier répertoire) est également faite de façon unique. Un répertoire privé par client limite la restriction dans les noms de fichiers aux seuls fichiers du client.

Nous allons voir maintenant comment un bloc d'un fichier est échangé entre un programme et le disque public, au moyen de la procédure utilisant les opérateurs programmés.

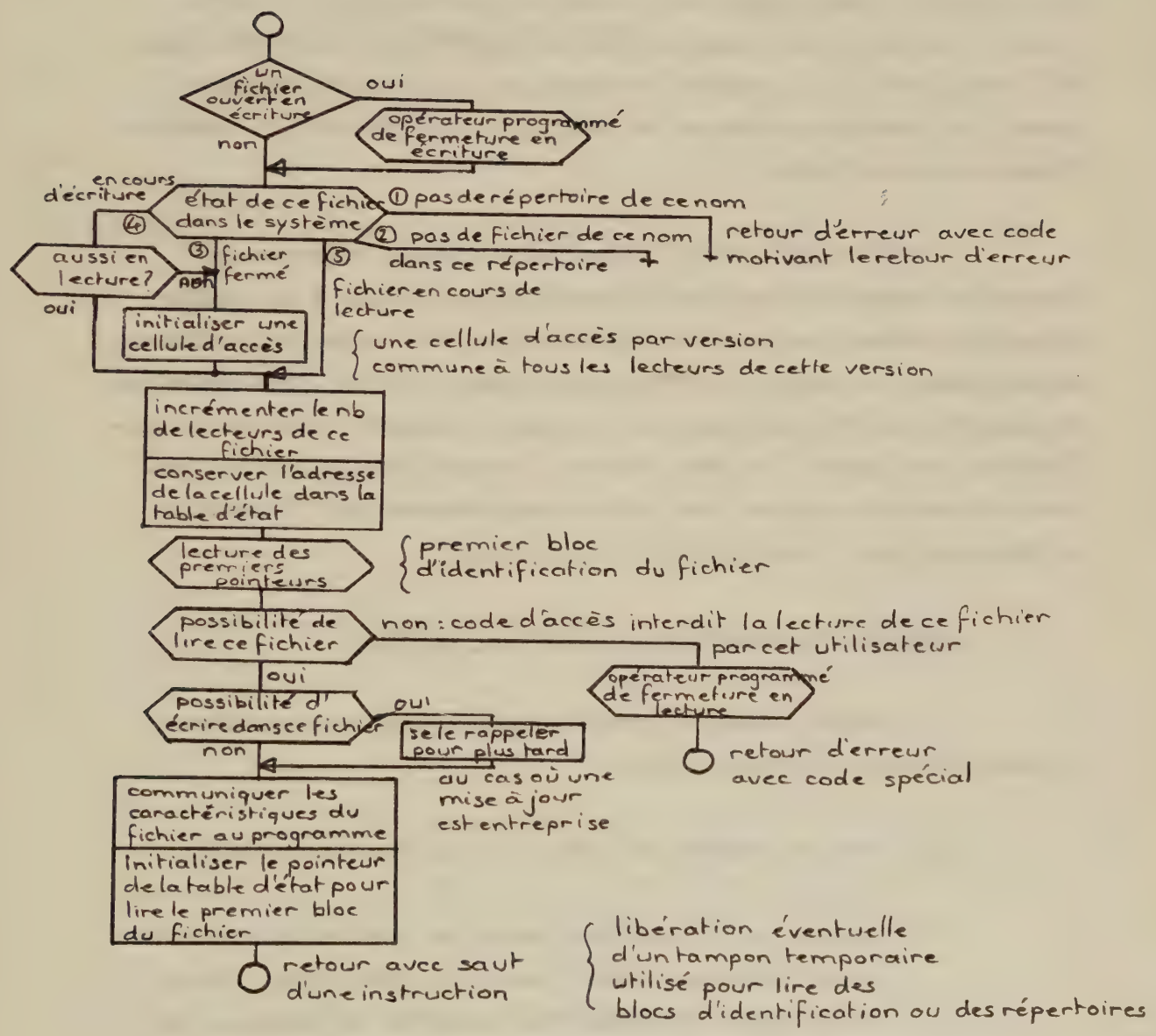
3 - Initialisation de l'équipement DISQUE (ou tout autre nom d'équipement ayant fait au préalable l'objet d'un ordre d'appellation symbolique)

L'opérateur programmé d'initialisation ne conduit pas à une routine spécifique sauf dans le cas du disque pour lequel chaque initialisation entraîne la construction d'une nouvelle table d'état (de même que chaque réservation avec appellation symbolique) à partir d'un modèle et insertion dans la chaîne des tables d'état. Cette table est détruite à la libération de l'équipement.

Les tables d'états sont construites dans une zone de mémoire gérée dynamiquement par le superviseur, lorsque cette zone est pleine, le disque est considéré comme inexistant et le retour approprié est donné à l'opérateur programmé.

4 - Ouverture d'un fichier de disque en lecture partageable

Lors de l'entrée dans la routine spécifique de cette opération des disques, on est assuré de la fermeture préalable d'un fichier déjà ouvert en lecture sur le même canal programmé.



Si un fichier est déjà ouvert en écriture sur cette même table d'équipement, il est fermé automatiquement un canal programmé n'étant pas prévu sur le disque, pour lire et écrire 2 fichiers différents, il sera toutefois possible d'écrire le fichier ouvert au préalable en lecture : il y a alors mise à jour.

Pour lire et réécrire un fichier, il faut nécessairement utiliser deux canaux programmés différents conduisant à deux tables d'état d'équipement différentes et quasiment indépendantes. L'état d'un fichier identifiable par le nom du répertoire et le nom du fichier (2,5 mots) est mémorisé dans le superviseur dans une cellule de 4 mots créée à la première ouverture en lecture (et une autre créée à la première ouverture en écriture) et détruite à la dernière fermeture de ce fichier.

Cette cellule porte l'état de ce fichier (écriture ou lecture et nombre de lecteurs, destruction du fichier à la dernière fermeture lecture par exemple). Les cellules de 4 mots sont chaînées bâties comme les tables d'état de disque dans la zone libre dynamique du superviseur. Ces cellules sont les seuls éléments indiquant l'utilisation simultanée d'un même fichier, partagé entre plusieurs travaux indépendants ou non. Ces indications communes sont reprises à la fermeture des fichiers ouverts sur disque.

5 - <u>Ouverture d'un fichier de disque en écriture</u>

Le fichier concerné identifié par le nom du répertoire et le nom du fichier (2,5 mots) peut :

a) - ne pas exister du tout : il s'agit d'une création,

b) - exister et ne pas être ouvert en lecture pour le même canal programmé (même travail et même table d'état) : il s'agit d'une réécriture complète et distincte de l'ancienne version appelée à disparaître,

c) - exister et être en cours de lecture pour le même canal programmé (même travail et même table d'état d'équipement), il s'agit d'une mise à jour locale du fichier (certains blocs),

d) - être déjà ouvert en écriture, une seule écritue à la fois est possible, il faut avertir le programme par un retour d'erreur associé d'un code approprié, il suffit peut-être d'attendre un peu que le fichier soit refermé. (opérateur programmé de sommeil).

un fichier est ouvert en lecture ici ?

oui : il peut s'agir d'une mise à jour

non

état de ce fichier dans le système

① pas de répertoire de ce nom → code d'erreur

② pas de fichier à ce nom dans le répertoire

③ fichier fermé

④ fichier déjà en écriture

⑤ fichier en lecture

fichier aussi en écriture — oui

non

lecture des premiers pointeur et du code d'accès

droit d'écrire — non → retour avec code d'erreur

oui

préciser dans la cellule d'accès qu'il s'agit d'une écriture

préciser dans la future cellule d'accès qu'il s'agit d'une écriture et qu'il est déjà répertorié

oublier le bloc de tête d'identification

ré-exploration des cellules d'accès : nouvelle exploration car une attente a pu avoir lieu lors de la lecture des premiers pointeurs et les conditions ont pu changer sous l'action d'autres programmes

pas trouvé de référence à ce fichier

fichier maintenant en écriture — oui → retour avec code d'erreur

non

encore des cellules — non

oui — même fichier (lecture de l'ancienne version)

allocation d'un bloc pour écriture des nouveaux pointeurs et identification

création d'une cellule d'accès — { nouvelle version d'un fichier = nouvelle cellule d'accès (unique)

conserver l'adresse de la cellule dans la table d'état

mise en place du nom et des caractéristiques du fichier dans le bloc d'identification

initialiser le pointeur de la table d'état pour écrire le premier bloc — { premier pointeur dans le bloc d'identification pour le premier bloc du fichier à écrire

retour avec saut d'un instruction après ouverture d'une nouvelle version de fichier

droit d'écriture constaté — non → retour avec code d'erreur

oui

d'après cellule d'accès déjà en mise à jour — oui (par autre programme) → retour avec code d'erreur

non

indiquer : mise à jour

s'agit-il du même fichier — non → message et arrêt du travail (erreur de programmation)

oui

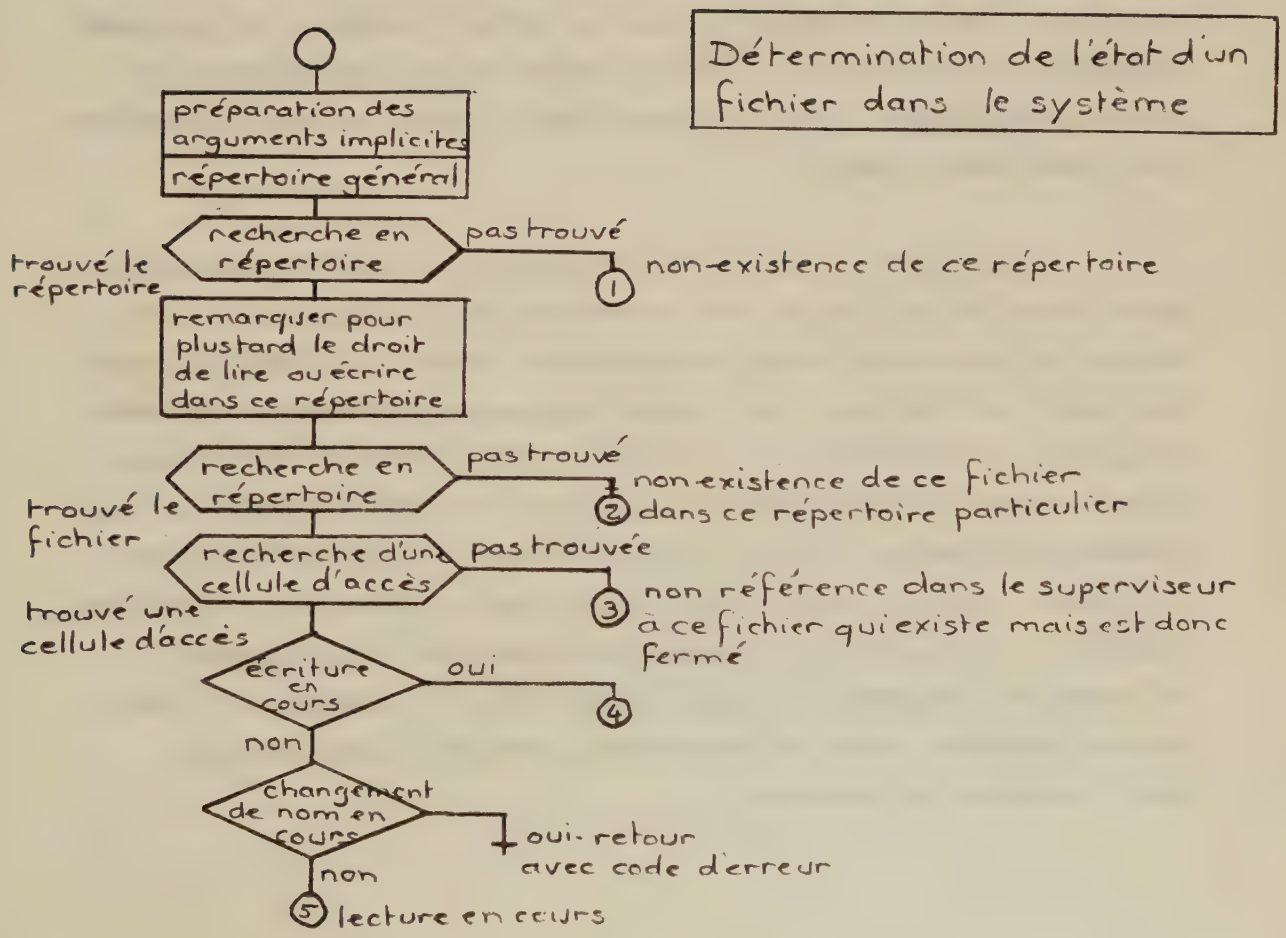
relecture de la tête du premier bloc d'identification. communiquer les caractéristiques du fichier au programme

retour avec saut d'une instruction après ouverture du fichier pour mise à jour

Ouverture d'un fichier en écriture routine spécifique des disques

Dans le système, il ne peut exister que 2 cellules d'accès pour un même fichier, l'un pour les lecteurs de l'ancienne version, l'autre pour le seul programme autorisé à réécrire ce fichier. En cas de mise à jour, il n'y a qu'une cellule puiqu'il n'y a qu'une seule version continuellement variable.

D'après l'algorithme d'exploration du système pour connaître l'état d'un fichier, les répertoires ont la priorité sur les cellules d'accès ce qui évite de se mettre à lire un fichier en cours de création ou de réécriture.



Les codes d'accès qui font partie des caractéristiques d'un fichier sont précisés, soit à la création du fichier, soit par un opérateur de changement de caractéristiques (nom). Ils permettent d'inhiber la lecture, l'écriture, le changement de code d'accès pour 3 catégories d'utilisateurs : le propriétaire du répertoire, les utilisateurs du système travaillant dans son équipe, (nom de répertoire "voisin" de celui-ci) et les autres utilisateurs, certaines combinaisons de ces 9 bits théoriquement indépendants, n'ont aucun sens et sont interprétées pour un autre usage (sauvegarde des fichiers à l'abandon du système par un utilisateur).

On peut au moyen de ce code restreindre le partage d'un fichier, soit à la lecture, soit à l'écriture, soit à soi-même, soit à une équipe, soit pas du tout.

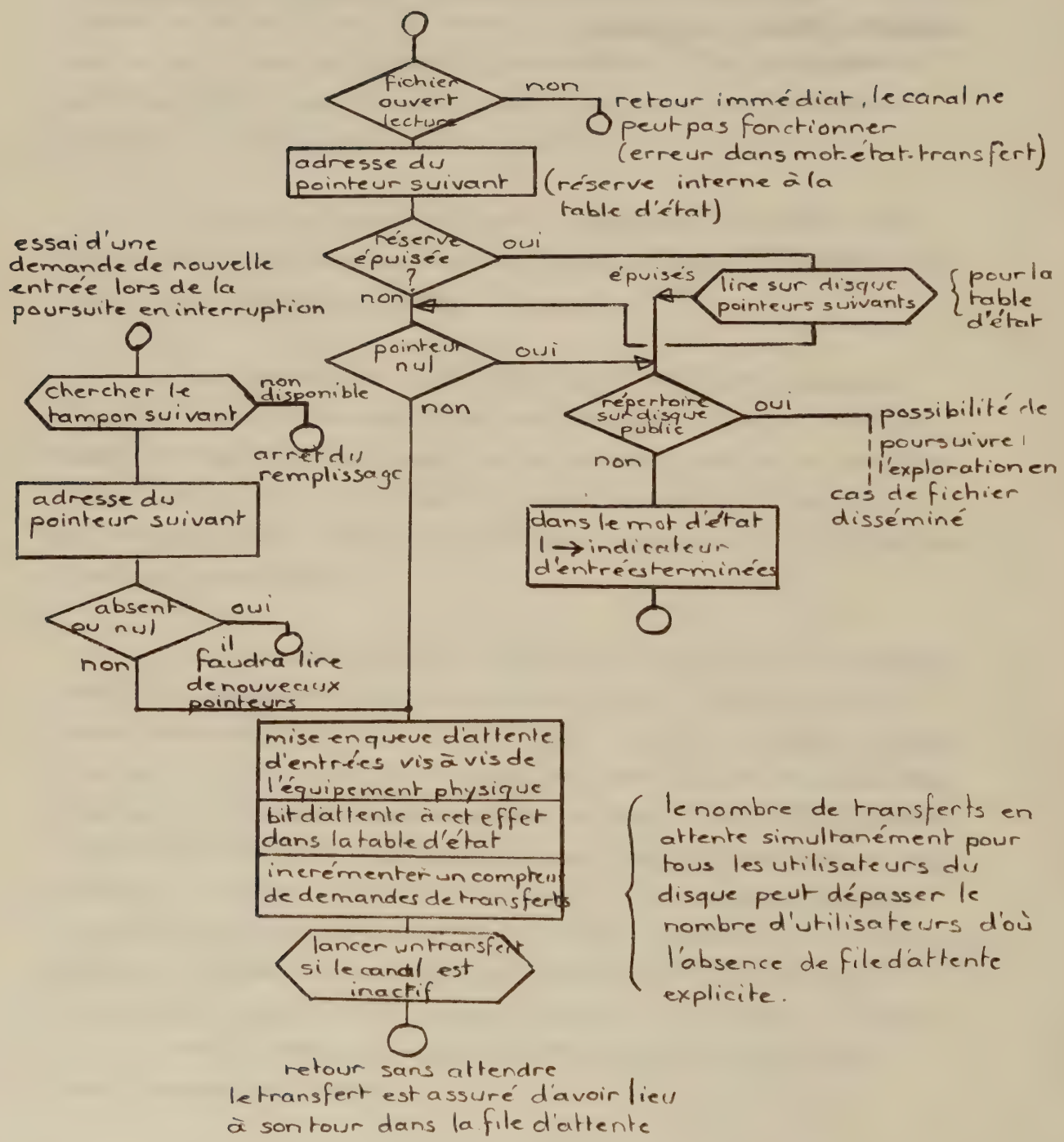
Les répertoires comme les fichiers ordinaires disposent d'un code d'accès interdisant leur modification, mais pas leur lecture, par d'autres programmeurs que le système lui-même, un programme travaillant en son nom ou l'administrateur du système détenteur du mot de passe adapté. (pour des opérations urgentes ou des éliminations d'erreurs inexpliquées).

Les fichiers sont bien partageables tant qu'un utilisateur privé propriétaire n'a pas mis de restrictions à l'usage de ses fichiers (sélectivement) mais le superviseur contrôle lui-même de façon sûre que 2 utilisateurs (ou 2 canaux programmés différents d'un même programme) ne cherchent pas à modifier en même temps un même fichier (même nom dans le même répertoire). Il est toujours possible, le cas échéant, d'attendre la fin de l'écriture par le programme autorisé le premier avant de poursuivre. C'est ce qui se passe pour le fichier général de comptabilité ou livre de bord manipulé inconsciemment au moyen des programmes d'entrée et de sortie du système, à l'exécution de certains ordres du superviseur, peut-être simultanément pour plusieurs utilisateurs.

Une simple estimation préalable du temps maximum de mise à jour permet la mise en sommeil des programmes avec réveils échelonnés jusqu'à ce que la non réussite persistante indique une anomalie par un message approprié et le rejet de l'utilisateur du système.

Ceci n'est qu'un cas particulier typique d'accès multiple à un même fichier partageable.

6 - Entrée du tampon suivant

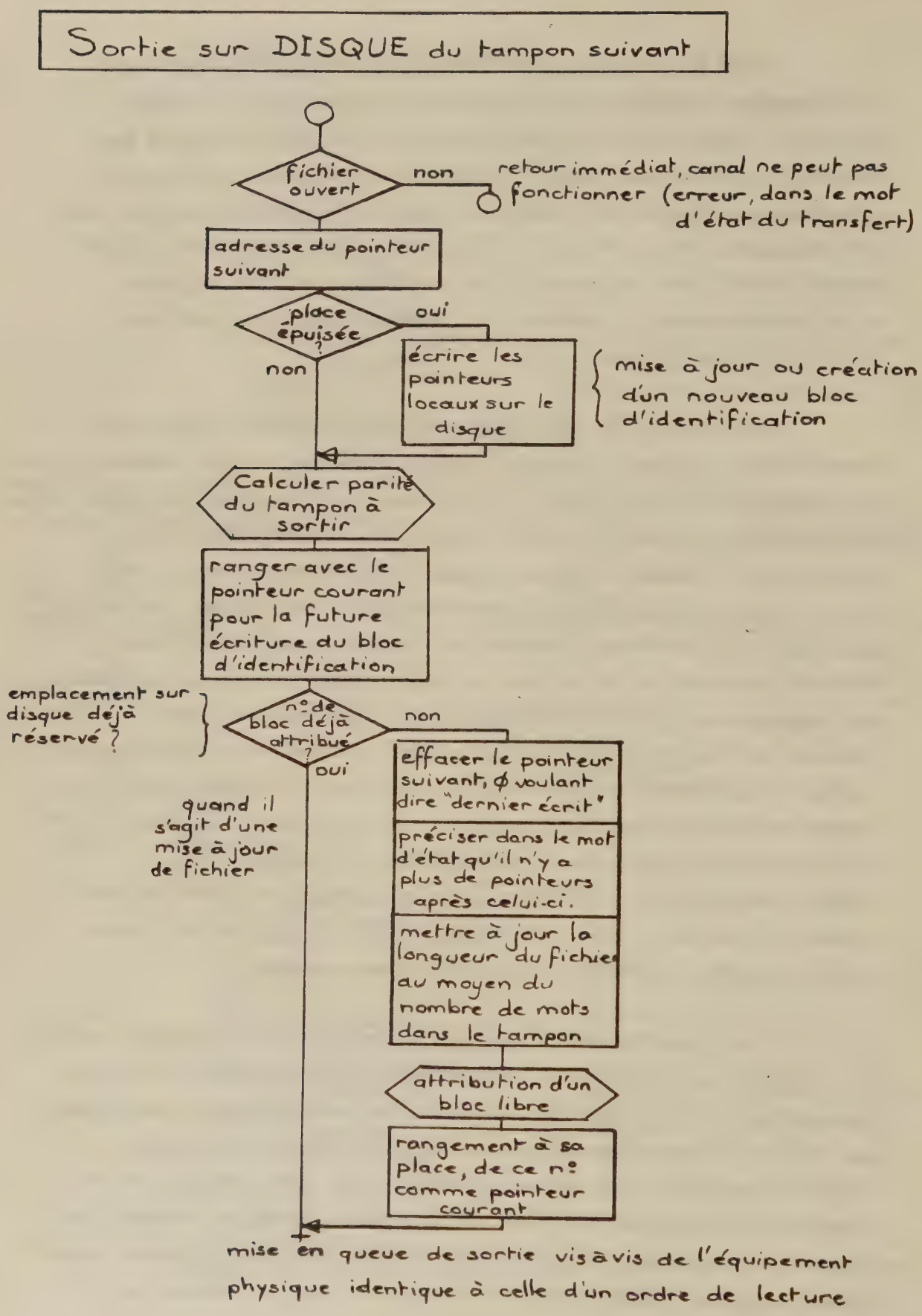


Lors de l'ouverture du fichier, une dizaine de pointeurs (les premiers) ont été lus et conservés en mémoire dans la table d'état de l'équipement qui sert de réserve. Lorsque la réserve est épuisée, le superviseur lance la lecture des suivants, en mode non prioritaire, c'est-à-dire lorsqu'un opérateur programmé d'entrée pour cet utilisateur et ce fichier (c'est-à-dire pour cette table d'équipement fictif) aura constaté le non remplissage du tampon présent ou du tampon suivant le présent, et pour cette raison, fait appel à la routine spécifique.

Une lecture de pointeurs obéit à une règle d'attente différente d'une lecture normale de tampon. Pour une lecture normale de tampons, il n'y a pas de file d'attente, car le nombre de transferts en attente peut être très variable et un même utilisateur peut en avoir plusieurs sur des canaux programmés différents, dans ce cas comme pour les ordres superviseur issus de consoles (en nombre peut être plus grand que le nombre de travaux simultanés prévus), c'est un compteur et un bit dans la table d'état de chaque équipement qui indiquent et permettent de retrouver les tâches en attente (lecture ou écriture) à la fin de la tâche en cours lors d'une interruption prioritaire. Les lectures (et écritures) décidées par le superviseur nécessitent leur achèvement pour que cette routine (ici saisie des pointeurs suivants), puisse continuer, donc le travail devra être mis en attente, ce qui exclut une opération semblable pour un autre canal programmé, la file d'attente doit alors être de longueur le nombre maximum de travaux prévus.

Les transferts à l'usage du superviseur sont toujours traités en priorité, car ils sont en général destinés à provoquer de nouveaux transferts (une fois lus les pointeurs, par exemple) donc plus pressés que les remplissages ou vidages de tampons dont les utilisateurs n'ont, en principe, pas un besoin immédiat ; ce principe qui est de favoriser toujours les éclosions de nouvelles tâches par rapport aux travaux en train dans le but de trouver plus facilement quelque chose à faire faire par l'unité centrale diminue le temps de réponse du système.

7 - Sortie du tampon suivant



Un seul fichier peut être ouvert à la fois, sur un canal programmé caractérisé essentiellement par la table d'état d'équipement associée et créée pour lui seul ; une ouverture en écriture d'un fichier déjà ouvert en lecture permet sa mise à jour sans recouvrement possible en cas d'erreur, puisque c'est la même place qui est utilisée pour récrire les blocs modifiés.

Dans le cas d'une création de nouvelle version ou de création d'un nouveau fichier ou d'allongement d'un fichier mis à jour, une nouvelle attribution de place sur le disque est nécessaire, elle est contrôlée par une table à usage exclusif du superviseur. Cette table est un fichier du disque, inaltérable par programme (protection contre toute écriture), dont la modification ne peut être faite que pour un seul utilisateur à la fois. Cet utilisateur est en général celui pour lequel travaille le superviseur à la suite d'un opérateur programmé, mais ce peut être aussi un travail en attente de fin d'entrée sortie sur disque, le transfert étant lui en attente dans la file d'attente des transferts pour le superviseur et ce transfert étant la lecture ou l'écriture d'un nouveau bloc de cette table d'occupation du disque. Cette attente pour mise à jour signifie en fait que la version en mémoire n'est pas correcte et est peut être en cours de modification par un transfet prioritaire.

Le même processus a lieu pour les mises à jour de répertoire par le superviseur.

Dès que la consultation a pu être faite après éventuellement une attente (signifiant, en fait, le passage à un autre travail, du bénéfice de l'unité centrale), le transfert dont tous les éléments sont prêts (adresse mémoire et adresse disque, longueur et emplacement réservé) est lancé ou mis en attente vis à vis de l'équipement physique, le contrôle étant de toute façon rendu immédiatement au programme appelant, ici la partie "sortie" du décodeur d'opérateurs programmés qui décidera de la suite des opérations.

De toute façon, l'équipement fictif est désormais dans l'état actif, cet état actif ne cessera que quand la demande de transfert aura été prise en compte, exécutée et vérifiée. Cette dernière tâche de lancement des transferts en attente et de vérification est assurée par la routine de traitement des interruptions venant des disques.

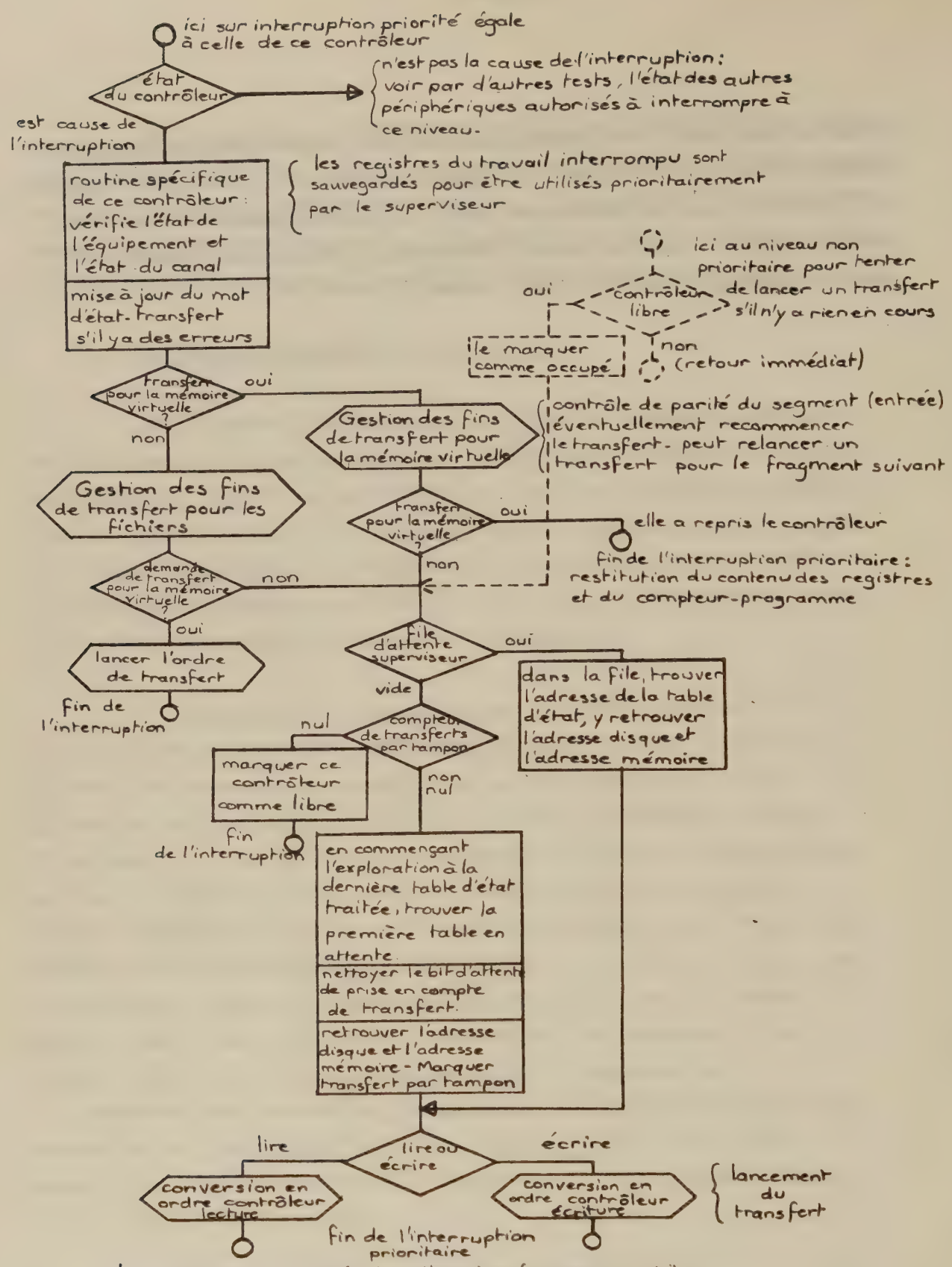
8 - Procédure d'interruption prioritaire

Un transfert physique est lancé par une routine spécifique de l'équipement adressé du point de vue logique ; un ordre de lecture ou d'écriture donné à un contrôleur de disque ou d'équipement similaire initialise le transfert. Lorsque celui-ci est terminé ou qu'une erreur a eu lieu, le contrôleur au moyen du système d'interruption prioritaire passe le contrôle de l'unité centrale à un sous-programme précis en vue de faire cesser la cause de l'interruption.

En pratique, à sa construction, le superviseur est prévu pour faire fonctionner les différents périphériques sur des niveaux de priorité précis et il considère qu'une priorité d'un certain niveau ne peut avoir pour cause que l'un des périphériques prévus. C'est pourquoi une interruption prioritaire d'un des 7 niveaux possibles va conduire par une série de tests à trouver à coup sûr, dans la chaîne des tests, le premier des équipements dont l'état physique a provoqué une demande d'interruption. Il n'est pas exclu que 2 équipements aient une demande déposée en même temps, le premier dans la chaîne des tests sera servi d'abord, ses registres d'état seront modifiés pour qu'il ne demande plus d'interruption, à la fin du traitement une instruction de cessation d'interruption avec retour au mode et à l'adresse du programme interrompu sera émise mais aussitôt le second périphérique verra à son tour sa demande d'interruption acceptée et traitée.

Les routines de traitement des interruptions venant de périphériques très différents mais prévus pour interrompre avec la même

priorité sont donc liées entre elles par des instructions de branchement lorsqu'un test de l'état de l'équipement prouve qu'il n'est pas l'auteur de l'interruption. La chaîne des tests ne peut que conduire à un test positif, dans le cas contraire (interruption parasite ou fugace) en bout de chaîne on tente de faire cesser l'interruption.



La dernière remontée de pile est prévue pour restituer le contenu des registres et passer le contrôle de nouveau au programme interrompu.

Traitement des interruptions prioritaires d'un contrôleur de disque(s)

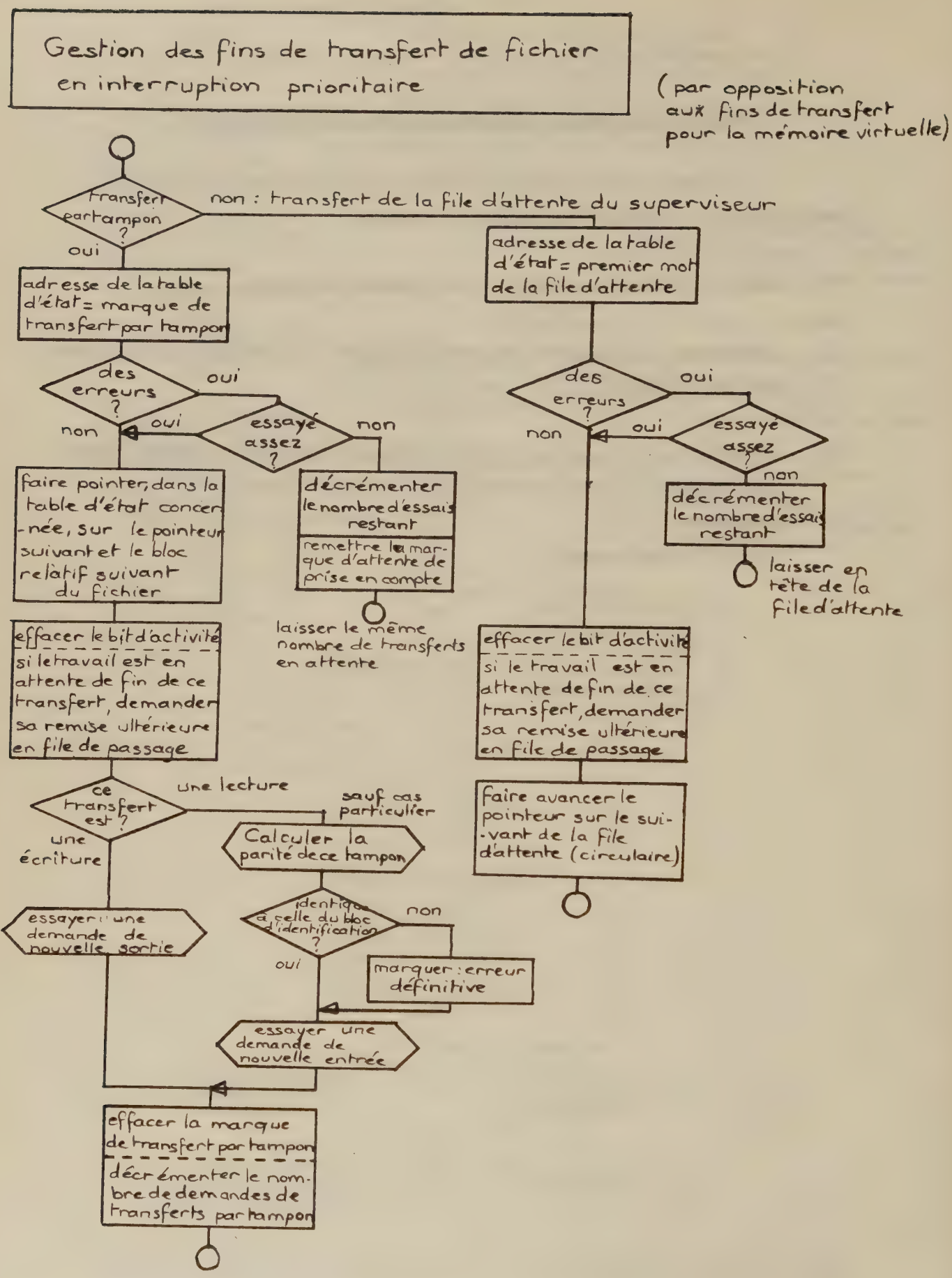
9 - Chaînage logique des ordres de transferts en attente

L'ordre dans lequel les attentes sont testées et prises en compte suffit à établir la priorité suivante :

1°/ - mémoire virtuelle gérée par son propre programme,

2°/ - transferts à l'usage du superviseur pour les opérateurs programmés sur les fichiers,

3°/ - transferts de blocs de fichiers séquentiellement en général dans des tampons des utilisateurs bien souvent d'avance à l'issu des programmes.
(voir l'organigramme).


Gestion des fins de transfert de fichier en interruption prioritaire
(par opposition aux fins de transfert pour la mémoire virtuelle)
transfert par tampon ?
non : transfert de la file d'attente du superviseur
oui
adresse de la table d'état = marque de transfert par tampon
adresse de la table d'état = premier mot de la file d'attente
des erreurs ?
oui
non
oui
essayé assez ?
non
faire pointer, dans la table d'état concernée, sur le pointeur suivant et le bloc relatif suivant du fichier
décrémenter le nombre d'essais restant
remettre la marque d'attente de prise en compte
laisser le même nombre de transferts en attente
des erreurs ?
oui
non
oui
essayé assez ?
non
décrémenter le nombre d'essais restant
laisser en tête de la file d'attente
effacer le bit d'activité
si le travail est en attente de fin de ce transfert, demander sa remise ultérieure en file de passage
effacer le bit d'activité
si le travail est en attente de fin de ce transfert, demander sa remise ultérieure en file de passage
faire avancer le pointeur sur le suivant de la file d'attente (circulaire)
ce transfert est ?
une lecture
sauf cas particulier
une écriture
Calculer la parité de ce tampon
identique à celle du bloc d'identification ?
non
oui
marquer : erreur définitive
essayer une demande de nouvelle sortie
essayer une demande de nouvelle entrée
effacer la marque de transfert par tampon
décrémenter le nombre de demandes de transferts par tampon

10 - Fermeture d'un fichier ouvert en lecture

Par cet opérateur programmé explicitement dans le programme de l'utilisateur ou plus souvent implicitement impliqué par un opérateur plus général (retour au superviseur, par exemple, en vue du passage dans la file d'attente des travaux stoppés), notification est donnée au système que le fichier consulté ne le sera plus par ce programme ou plus exactement par ce canal programmé.

Si, par ailleurs, une nouvelle version ou une destruction de l'unique version a été demandée dans le cadre du système, celle qui était en consultation a dû subsister pour ne pas pénaliser le ou les utilisateurs de cette ancienne version.

Comme pour les programmes partageables situés en mémoire, lors d'une mise en service d'une nouvelle version, le moyen d'accéder à l'ancienne version disparaît puisque le répertoire pointe sur le bloc d'identification de la nouvelle version.

Toutefois, cette ancienne version doit disparaître dès qu'elle n'aura plus de lecteurs.

Ses coordonnées sont conservées dans la table d'état d'équipement de chaque canal programmé qu'en fait la lecture, le dernier à travailler est chargé de la destruction du fichier.

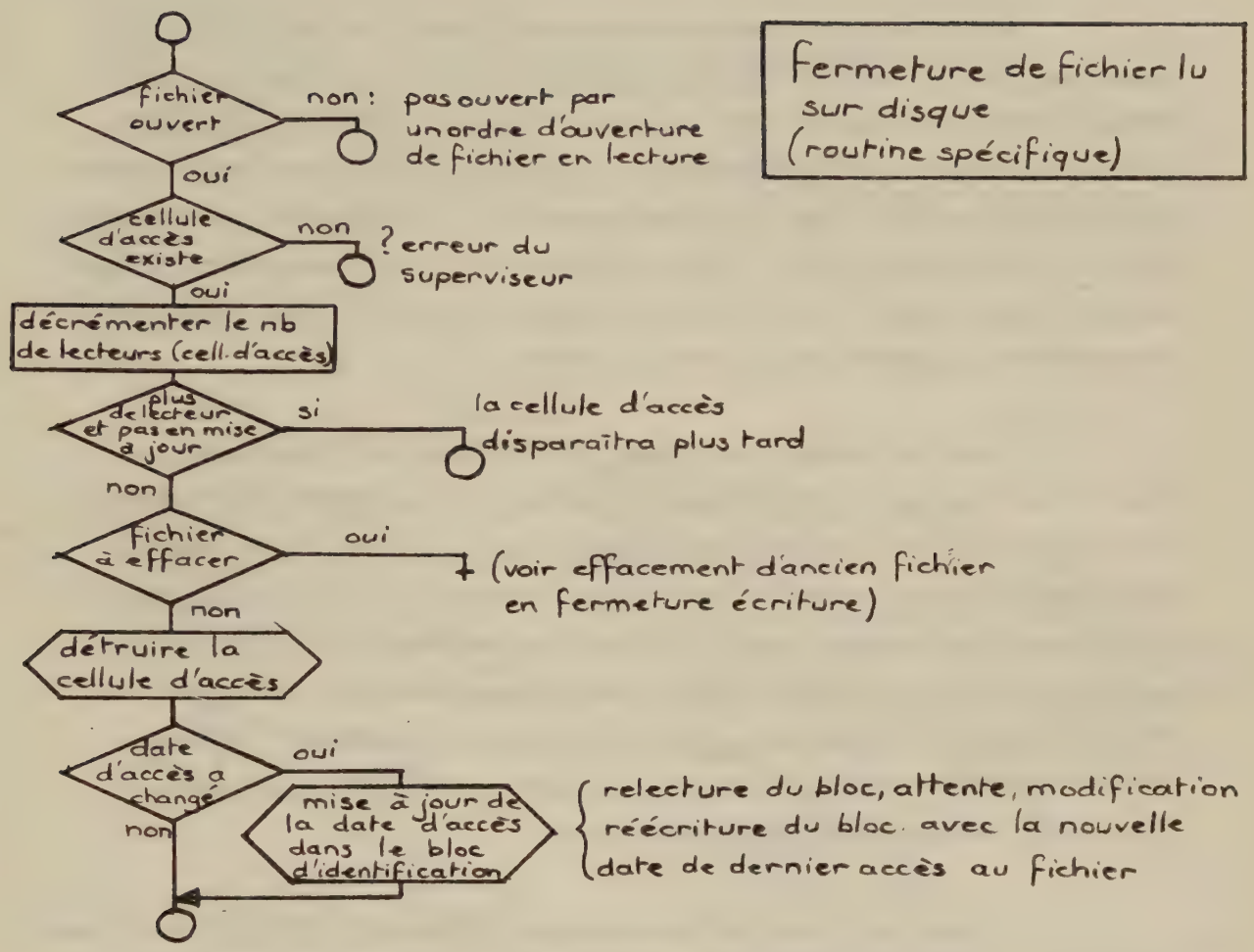
C'est à cet effet qu'un compte des lecteurs d'un fichier est conservé dans la cellule d'accès de ce fichier avec, en particulier, des indications comme la destruction nécessaire de ce fichier ou la constatation à l'ouverture que la date de dernier accès était la date du jour.

L'ordre des fermetures d'un fichier est d'abord la lecture puis l'écriture d'après la routine générale, mais il y a inhibition possible. C'est pour cette raison que la précaution est prise de ne

pas détruire une cellule d'accès s'il y a en même temps écriture et lecture, c'est-à-dire mise à jour de ce fichier.

C'est à la fermeture en écriture que la cellule disparaîtra s'il n'y a plus de lecteurs. Lorsqu'un fichier est en modification (unique) les lecteurs profitent d'une version toujours à jour, mais avec des risques d'indication d'erreur, les pointeurs utilisés contiennent des contrôles de parité périmés pour certains blocs réécrits dans l'intervalle entre la lecture des pointeurs et la lecture des blocs eux-mêmes.

Il paraît très difficile de se prévenir contre ce risque inhérent à l'asynchronisme propre au processus de partage, autrement qu'en interdisant une lecture simultanée d'un fichier en mise à jour. La possibilité, avec le risque connu, de ce genre d'erreur a été maintenue dans la mesure où elle ne met pas en cause la sécurité du système.

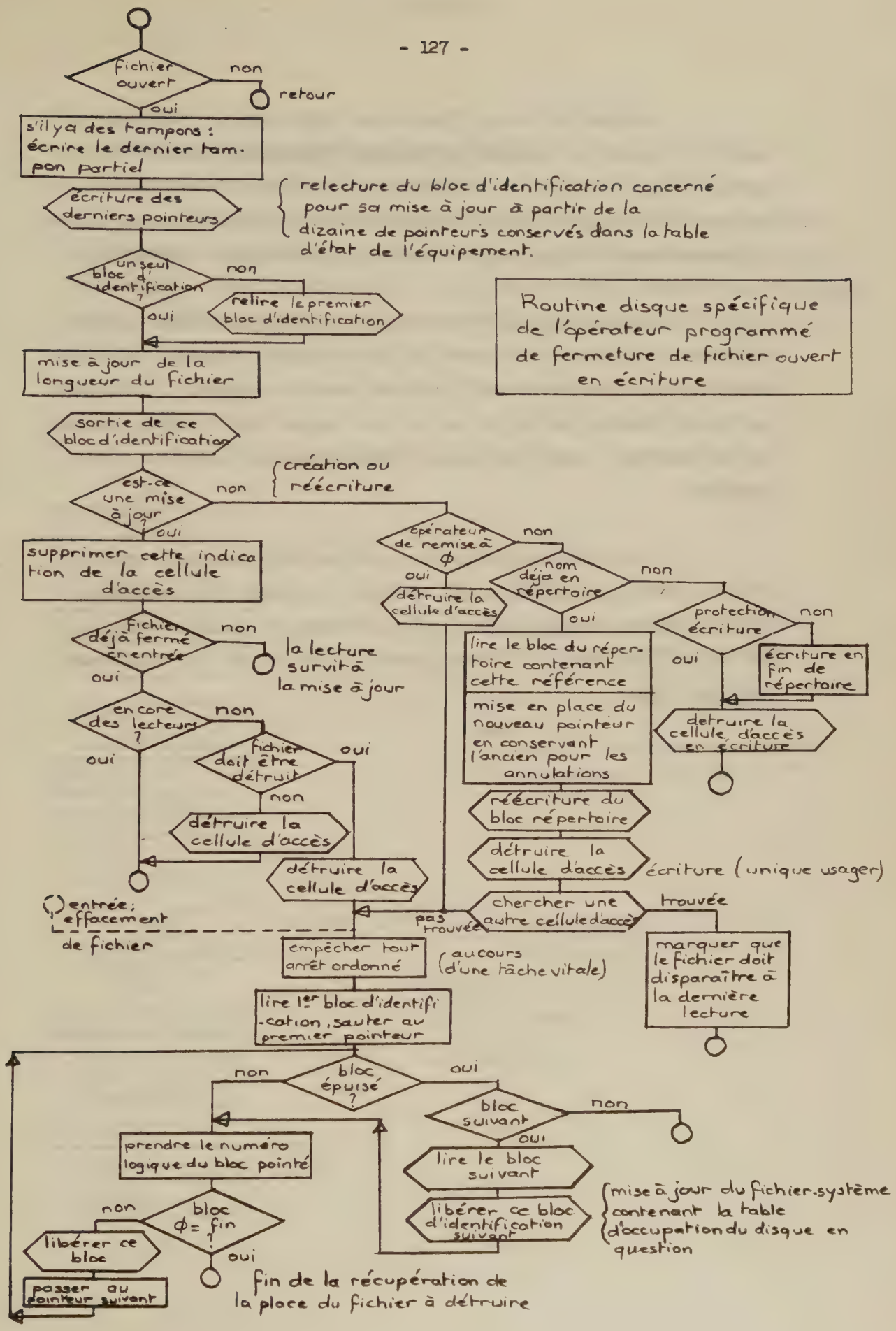

Fermeture de fichier lu sur disque (routine spécifique)
fichier ouvert
non : pas ouvert par un ordre d'ouverture de fichier en lecture
oui
cellule d'accès existe
non
? erreur du superviseur
oui
décrémenter le nb de lecteurs (cell. d'accès)
plus de lecteur et pas en mise à jour
si
la cellule d'accès disparaîtra plus tard
non
fichier à effacer
oui
(voir effacement d'ancien fichier en fermeture écriture)
non
détruire la cellule d'accès
date d'accès a changé
oui
mise à jour de la date d'accès dans le bloc d'identification
relecture du bloc, attente, modification
réécriture du bloc avec la nouvelle
date de dernier accès au fichier
non

11 - Fermeture d'un fichier ouvert en écriture

L'écriture du dernier tampon est d'abord assurée et la longueur du fichier calculée au fur et à mesure de son écriture, dans la table d'état, est rangée dans le bloc d'identification. S'il s'agit d'une mise à jour, le complément de ce qui ne fut pas fait en fermeture du côté lecture doit être fait maintenant si l'on était le dernier lecteur du fichier mis à jour ; éventuellement, en tant que lecteur d'une version complètement récrite par ailleurs, la version mise à jour doit disparaître au profit de la version écrite.

C'est un risque lors d'une mise à jour de fichier que la mise à jour ne serve à rien, sauf le temps de l'ouverture du fichier, une nouvelle version étant mise en place par ailleurs. Pour être sûr que la mise à jour subsiste, la solution consiste à essayer soi-même l'écriture sur un autre canal programmé en maintenant le fichier ouvert en écriture condamnant toute tentative extérieure similaire, la mise à jour étant terminée, le fichier refermé, remettre à zéro sa propre tentative de réécriture pour laisser subsister malgré tout l'ancienne version maintenant mise àjour.

Lorsqu'une fermeture en écriture a lieu sous l'effet de l'opérateur programmé de remise à zéro des entrées-sorties en cours, les tentatives d'écriture de nouveaux fichiers non encore fermés sont effacées du système de manière à laisser subsister les anciennes versions en particulier, en cas d'erreur pendant la création des nouvelles.
Un cas typique est celui d'un programme en mise au point, qui boucle, que l'on arrête par ordre au superviseur et que l'on remplace par appel d'un autre programme, tous les fichiers en cours d'écriture sont détruits et rien ne subsiste des tentatives d'écriture sur disque.
Ceci soulage beaucoup l'utilisateur du système et limite la gêne que constitue une occupation inutile de place sur disque par des fichiers sans valeur.


fichier ouvert
non
retour
oui
s'il y a des tampons : écrire le dernier tampon partiel
écriture des derniers pointeurs
relecture du bloc d'identification concerné pour sa mise à jour à partir de la dizaine de pointeurs conservés dans la table d'état de l'équipement.
un seul bloc d'identification ?
non
relire le premier bloc d'identification
oui
mise à jour de la longueur du fichier
sortie de ce bloc d'identification
Routine disque spécifique de l'opérateur programmé de fermeture de fichier ouvert en écriture
est-ce une mise à jour ?
non
création ou réécriture
oui
supprimer cette indication de la cellule d'accès
fichier déjà fermé en entrée
non
la lecture survit à la mise à jour
oui
encore des lecteurs ?
non
oui
fichier doit être détruit
oui
non
détruire la cellule d'accès
détruire la cellule d'accès
entrée : effacement de fichier
opérateur de remise à Φ
non
oui
détruire la cellule d'accès
nom déjà en répertoire
non
oui
lire le bloc du répertoire contenant cette référence
mise en place du nouveau pointeur en conservant l'ancien pour les annulations
réécriture du bloc répertoire
détruire la cellule d'accès
écriture (unique usager)
protection écriture
non
oui
écriture en fin de répertoire
détruire la cellule d'accès en écriture
chercher une autre cellule d'accès
pas trouvée
trouvée
empêcher tout arrêt ordonné
au cours (d'une tâche vitale)
marquer que le fichier doit disparaître à la dernière lecture
lire 1er bloc d'identification, sauter au premier pointeur
bloc épuisé ?
non
oui
bloc suivant
non
oui
lire le bloc suivant
libérer ce bloc d'identification suivant
mise à jour du fichier-système contenant la table d'occupation du disque en question
prendre le numéro logique du bloc pointé
bloc Φ = fin ?
non
oui
libérer ce bloc
passer au pointeur suivant
fin de la récupération de la place du fichier à détruire

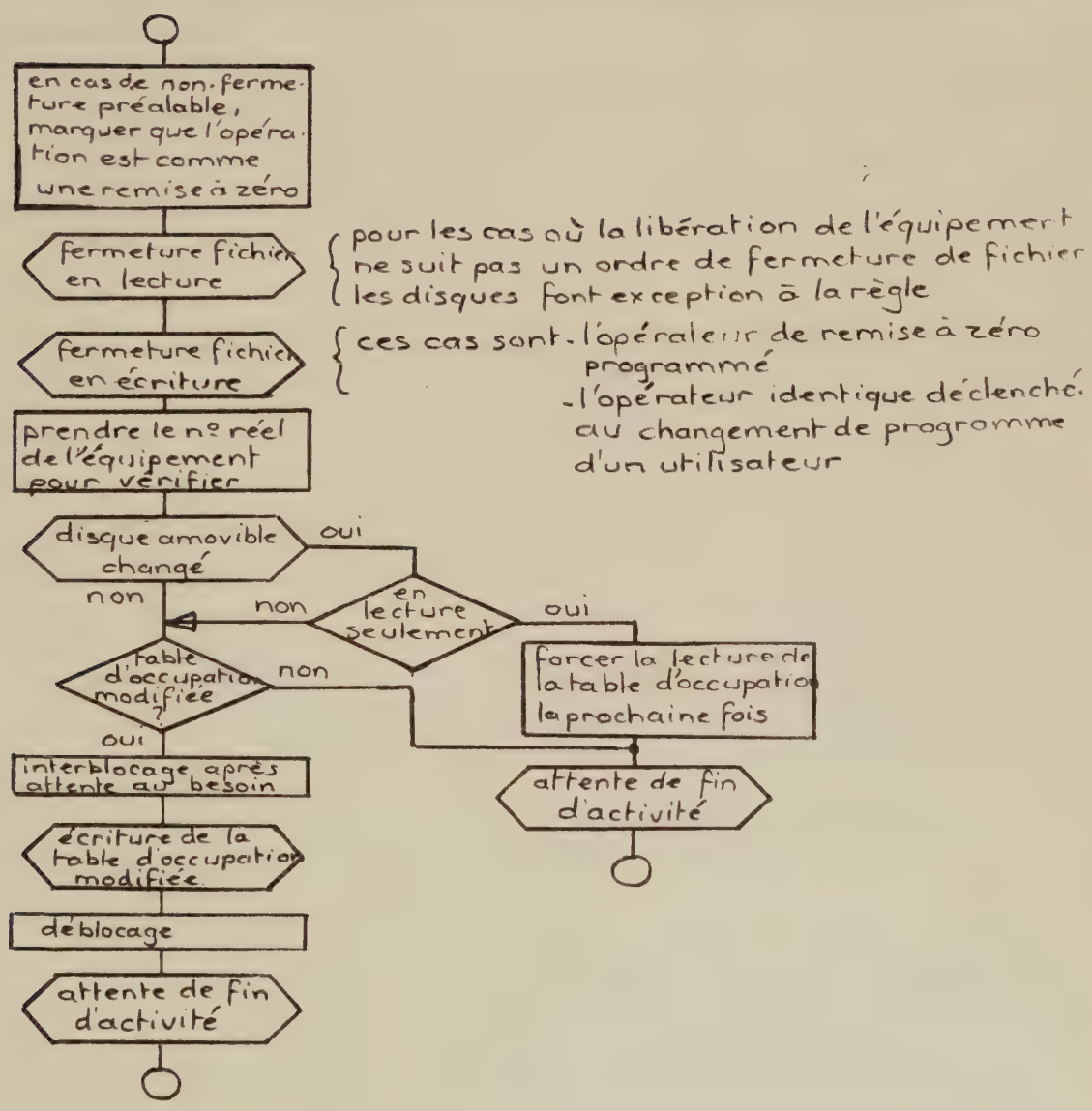
Lorsque le nom ne figure pas déjà dans le répertoire, il faut l'y insérer et récrire le fichier répertoire. Lorsqu'il y figure, il faut modifier le pointeur sur le bloc d'identification et préparer la destruction de l'ancienne version, soit tout de suite si personne ne la lit, soit à la fin de la dernière lecture.

La destruction d'une vieille version est faite par le superviseur à partir du numéro du bloc d'identification lu et détruit et tous les blocs qu'il pointe jusqu'au premier Ø indiquant la fin du fichier ou la fin de ce bloc auquel cas le même processus se poursuit pour les blocs d'identification suivants s'ils existent.

La destruction d'un bloc est en fait la simple mise à jour du fichier du système qui contient la table d'occupation. Il y a attente éventuellement pour modifier ce fichier comme il a été vu à l'écriture sur disque quand on désire se voir attribuer un bloc libre.

12 - Libération de l'équipement disque

Elle a pour but de forcer ce qui n'a pas lieu pour d'autres équipements (ou qui est une fonction très spécifique) à savoir l'effacement et la récupération de la place des fichiers encore ouverts mais dont la fermeture n'a pas été menée à bien (comme pour l'opération de remise à zéro).

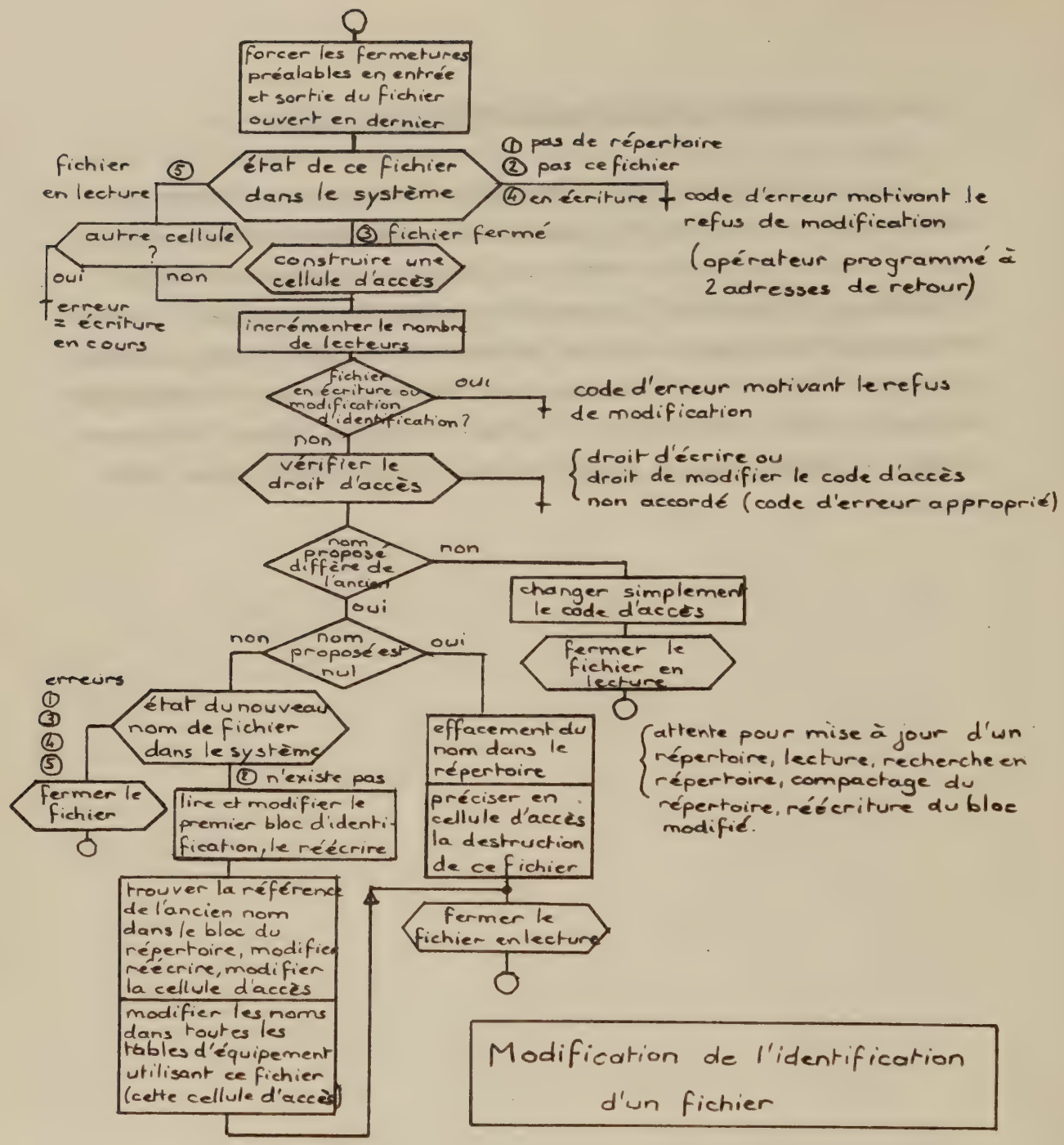


Le but final est surtout la réécriture de la table d'occupation du disque dont la version en mémoire diffère de la version sur le support extérieur. L'accès à cette table est toujours fait par file d'attente comme il a été expliqué plus haut.

La routine générale de libération d'équipement assure en outre et plus tard la destruction de la table d'état de l'équipement libéré dans le cas particulier des disques.

13 - Modification de l'identification d'un fichier

Un opérateur valable pour tous les équipements permet de changer soit le nom d'un fichier, soit la date de création, soit le code d'accès d'un fichier donné sans toucher au contenu du fichier lui-même. L'attribution d'un nom nul signifie par convention l'effacement pur et simple de ce fichier sélectionné obligatoirement au préalable par l'opérateur d'ouverture de fichier en lecture. Le changement de nom doit être répercuté dans toutes les tables d'état d'équipement lisant ce fichier pour la fermeture ou d'autres modifications ultérieures.



Modification de l'identification d'un fichier

CHAPITRE 4

ETUDE D'UN SYSTEME DE PAGINATION PROGRAMME DANS UNE STRUCTURE DE PARTAGE DE TEMPS

I - PRESENTATION DU PROBLEME

Si l'existence d'une mémoire virtuelle et d'un mécanisme d'échange avec la mémoire réelle peut assurer à chaque utilisateur du système le bénéfice de la totalité de la mémoire réelle disponible, il ne lui est toutefois pas possible de faire exécuter des programmes plus longs que le maximum disponible.

En fait, l'exécution proprement dite nécessite rarement un tel volume d'instructions-machine mais plus souvent de la place pour des tables ou des tableaux adressés sélectivement. Cet adressage sélectif provient bien souvent des difficultés d'adresser sélectivement les mêmes données rangées sous forme de fichier modifiable sur un équipement périphérique, de préférence à la mémoire réelle.

Le temps d'accès à un fichier est évidemment notoirement plus long (plusieurs dizaines de millisecondes au lieu de quelques microsecondes) mais les travaux à gérer par le superviseur sont beaucoup plus petits et les échanges avec la mémoire virtuelle sont en conséquence moins fréquents et moins volumineux.

Dans un système très chargé dont les travaux gérés occupent chacun plus de la moitié de la zone mémoire disponible, l'unité centrale reste inoccupée (travail nul ou d'attente), alors que des travaux situés en mémoire virtuelle sont en cours de transfert mais

auraient pu tourner s'ils avaient été en mémoire réelle. Il y a donc perte de temps au nom de l'équité entre gros travaux individuels.

Pendant le temps passé à ces échanges, de petits travaux peuvent très bien trouver leur place et profiter de l'unité centrale.

Tout système proposé doit amener le minimum de modifications au système déjà existant et être au maximum compatible pour utiliser les programmes de la bibliothèque présente et future.

Pour répondre au problème de la restriction du volume de programme présent en mémoire réelle rapide, plusieurs solutions ont déjà été proposées et utilisées :

- le chaînage déjà prévu pour les programmes FORTRAN,

- les appels sans retour d'un programme à un autre,

- des solutions non implantées dans ce système :

  le recouvrement ou la segmentation avec ou sans translation du code avant exécution.

L'inconvénient majeur de toutes ces méthodes est la préparation spécifique des segments et des morceaux de programme demandant à l'utilisateur un effort supplémentaire de connaissance du système.

Pour simuler une mémoire supérieure banalisée comme la mémoire offerte normalement par le système à un programme utilisateur, l'idée la plus simple fut de l'implanter en totalité sur une mémoire lente non adressable sélectivement (disque rapide), de la découper en pages de longueur fixe et de ne placer à un moment donné, en mémoire réelle, que la page de travail contenant les instructions en cours d'exécution.

2 - PRESENTATION DU PROGRAMME AVANT L'EXECUTION

Il s'agit d'une suite de pages consécutives situées en mémoire lente (sous forme de fichier sur disque d'accès rapide si possible), cette suite constitue le segment complet et unique du programme.

3 - CONTROLE DU PROGRAMME PAGINE : MONITEUR

Le contrôle repose sur un programme extérieur au programme paginé. Il s'agit d'un véritable moniteur dont le rôle essentiel est de vérifier le bon déroulement du programme paginé.

Le moniteur a d'abord le contrôle pour apprendre de la console de l'utilisateur le nom de fichier du programme paginé et son emplacement en mémoire lente (nom du répertoire). Le moniteur effectue un certain nombre d'initialisations, au moyen de la taille de chaque page, il traduit certaines adresses à l'usage du superviseur, il en modifie d'autres pour le programme lui-même, il charge en mémoire réelle la page contenant l'instruction de départ, met en place les mécanismes d'interruption et lance l'exécution de l'instruction de départ.

4 - LES CAUSES D'INTERRUPTIONS DE PROGRAMME

Le raisonnement et toutes les explications concernant le système de pagination se situent à un niveau très inférieur à celui où le partage de temps est effectué. On peut faire facilement abstraction de ce dernier puisqu'il va s'agir le plus souvent d'interactions entre 2 parties du programme de l'utilisateur : son moniteur et son programme paginé.

Cette abstraction est possible grâce à la transparence de l'ordonnancement des tâches à l'égard des travaux gérés qui se traduit par l'apparence de simultanéité et de continuité.

Les causes d'interruption du programme paginé sont en fait identiques à celles qui touchent un programme normal, les processus physiques étant forcément les mêmes. Compte tenu de la connaissance de l'organisation en page des programmes, les actions entreprises lors des interruptions seront particulières et sous le contrôle du moniteur seul.

Ces causes d'interruption sont :

a) - l'accès à une adresse interdite (violation de mémoire)

C'est qu'une tentative a été faite d'accéder à une adresse située hors des limites assignées par le superviseur au programme de l'usager (ici c'est sa page de travail seule présente en mémoire réelle, le moniteur est lui au-dessus de tout soupçon !). Cette tentative provoque en général la fin prématurée de l'instruction en cours et une demande d'interruption très prioritaire (niveau d'interruption accordé à l'unité centrale, par exemple, pour la panne de secteur ou l'erreur de parité). Cette interruption traitée par le superviseur peut provoquer le branchement à une adresse indiquée par le programme-usager dans sa propre zone, s'il en a fait la demande préalable.

Dans ce cas, l'interruption prioritaire cesse, au profit du programme de l'usager, se poursuivant à l'adresse de branchement indiquée.

Si aucune demande préalable n'a été faite, un message d'erreur est imprimé sur la console de l'utilisateur après arrêt de son programme (l'arrêt pour erreur est une tâche non différée, exécutée au niveau de l'horloge pour lancer immédiatement un nouveau travail). Ici, la demande préalable conduit à un traitement par la partie moniteur de pagination de la zone mémoire attribuée à l'usager. Le moniteur devra effectuer ce que n'a pu faire directement l'unité centrale.

b) - <u>impossibilité d'exécuter une instruction</u> (non câblée)

- les codes réservés à l'usager ne sont pas traités par le moniteur mais renvoyés à l'adresse du programme paginé,

- les codes normalement réservés à la communication avec le superviseur devraient être exécutés par lui, mais il a été modifié pour que le moniteur décide au préalable de la suite àdonner à chaque opérateur programmé. Pour ne pas trop modifier le superviseur, le contrôle a priori et la préparation de certains arguments en forme et lieu acceptables par lui sont effectués par le moniteur,

- les codes d'instructions non existantes, qui normalement ne devraient pas figurer dans les programmes (et provoquent l'impression d'un message d'erreur sur la console de l'usager) ont été utilisés pour remplacer certains codes mal adaptés au système de pagination (perte fatale d'information en cas de violation de mémoire). Ceci revient à faire simuler certaines instructions-machine en fonction de la pagination ; pour cela, il faut utiliser à l'assemblage des codes non exécutables pour que l'instruction devienne un opérateur programmé.

5 - <u>IMPLANTATIONS DU PROGRAMME ET DU MONITEUR</u>

Rappel sur le segment d'un utilisateur du système :

. les adresses Ø à 17 ne sont pas translatées (registres rapides)

. les adresses 2Ø à n x 2ØØØ-1 sont translatées : segment bas

. les adresses 2Ø à 137 servant à communiquer avec le superviseur

. les adresses 14Ø à n x 2ØØØ-1 étant librement utilisées

. les adresses 4ØØØØØ à 4ØØØØØ-1+m * 2ØØØ sont translatées au moyen du second registre de base (modulo 256 K) : segment haut peut être protégé contre l'écriture (et contre la lecture).

a) - <u>définition d'une page</u>

Une page est une zone continue autoadressable par les instructions normales de la machine. Il est possible depuis la page située en mémoire réelle d'adresser une page différente par un moyen extérieur au programme de l'usager (câblage ou, dans notre cas, moniteur).

Il est d'autre part indispensable pour la compatibilité des programmes avec ou sans le système de pagination, que les calculs sur les adresses (longueur d'un tableau par exemple) soient possibles comme en exécution normale.

L'idée retenue pour le découpage des programmes a été d'avoir des pages de taille multiple de 1024 mots dont la borne supérieure peut être celle du mécanisme de protection et dont :

- les adresses sont <u>réelles</u> (relatives au début de page) pour les positions situées dans la même page,

- les adresses sont absolues (relatives au début de fichier) pour les positions situées dans les autres pages.

Ceci répond au besoin :

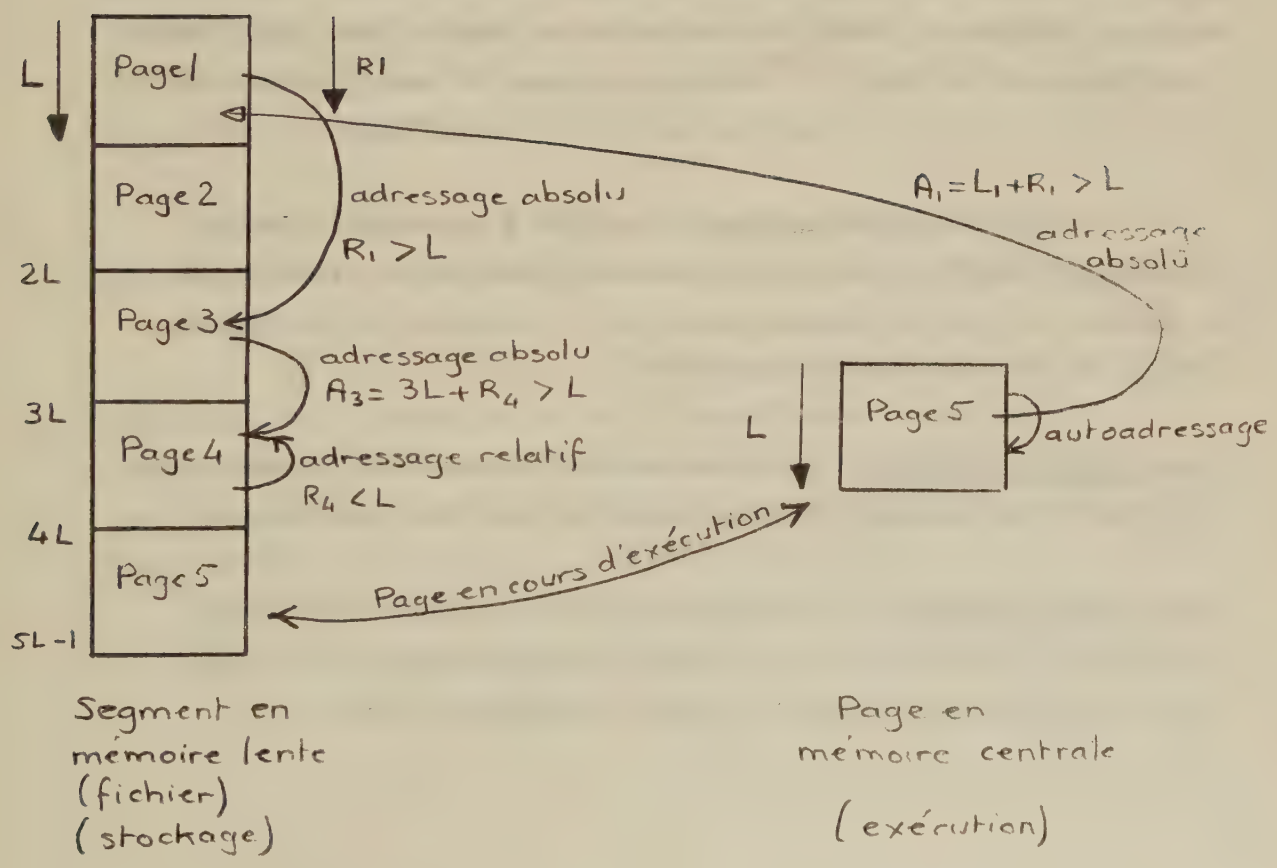
a) - d'autoadressage dans une même page,

b) - de compatibilité car seule la valeur de l'adresse va permettre la distinction entre <u>réel</u> et <u>absolu</u>.

La reconnaissance la plus simple, si les pages ont toutes la longueur L, est tout simplement :

adresse $R < L$ — indique un déplacement dans la page en cours, toléré par le mécanisme de protection et exécutable sans problème par l'unité centrale comme un programme normal.

adresse $R \geqslant L$ — indique un déplacement absolu dans le segment complet (fichier) constitué d'un certain nombre de pages, il n'est pas toléré par le mécanisme de protection et l'exécution d'une instruction utilisant un pareil déplacement est du ressort du moniteur (module Unité Centrale Programmée)



Pour que même la première page puisse être adressée par les autres en absolu, il faut que son adresse soit supérieure à L, longueur d'une page, ce qui revient puisque les adresses absolues des positions de la page n sont comprises entre nL et (n+1)L-1, à dire que la page Ø n'existe pas, cette page Ø étant la page en mémoire centrale, servant à l'exécution et dont le contenu est variable au cours du déroulement du programme. Chaque page devient à son tour page Ø pour laquelle adresses absolues et adresses relatives coïncident, lorsqu'elle est appelée pour exécution en mémoire centrale : l'autoadressage est l'adressage de la page Ø.

b) - propriété de translatabilité

Tout ce qui vient d'être dit repose sur la notion d'adresse, cette notion peut être comprise comme un déplacement relativement à une valeur connue ; en fait, tout y est translatable de cette même quantité (adresse de tête), les adresses relatives comme les adresses absolues, simplement l'adresse absolue de la page 1 n'est plus L mais L_0 = T (adresse de tête) + L longueur de page, la page Ø allant de T à T + L - 1 = L_0 - 1 valeur du registre de protection.

Cette propriété de translatabilité a permis de résoudre l'impossibilité d'utiliser le système de protection et translation du segment bas pour le bornage supérieur et inférieur de la page de travail : il y a à cette impossibilité :

- des raisons physiques : les adresses Ø à 17 réservées aux registres rapides et non au code des programmes, ne sont pas translatées,

- des raisons logiques les adresses 2Ø à 137 sont translatées mais réservées au superviseur ou au moins partagées avec lui et donc inutilisables pour ranger les codes d'une page en cours.

Ces restrictions auraient conduit à des trous dans l'adressage rendant impossible tout calcul simple sur les adresses, par les programmes paginés s'il avait fallu interdire dans chaque page les adresses Ø à 137.

La solution à ce problème, besoin d'avoir des pages pleines bornées de Ø à L et non de 14Ø à L, fut apportée par l'utilisation du second registre de translation et de sa protection de zone, multiple de 1 K. Son utilisation suppose simplement que les adresses seront supérieures à 4ØØØØØ, ce qui revient à translater l'adressage du début de segment et chaque page (adresse interne ou en page Ø) de Ø à 128. K (= 4ØØØØØ). Ceci ne présente pas de difficultés puisque tous les programmes sont compilés pour être en implantation variable.

Une page devient dans ces conditions une zone translatée haute, entre les adresses 4ØØØØØ et 4ØØØØØ - 1 + n * 2ØØØ (n ≠ Ø),
les adresses relatives (en page Ø) sont entre ces limites,
les adresses absolues sont supérieures à 4ØØØØØ - 1 + n * 2ØØØ.
On veillera à ce que la protection écriture ne soit jamais appliquée à la page de travail.

Les registres Ø à 2Ø sont toujours adressables de même qu'une éventuelle zone basse translatée (adresses 2Ø à 2ØØØ * m = - 1).

La longueur maximum d'un segment paginé est ainsi de 128. K ; moyennant une convention sur les adresses basses comprises entre 2ØØØ * m et 377777 (valeurs positives sur 18 bits), cette longueur peut être étendue facilement à 256. K (moins la page Ø, en fait non utilisable).

6 - POSITION DU MONITEUR DE PAGINATION

La page ayant été délimitée compte tenu des impératifs de bornage supérieur et inférieur et de continuité dans les pages, il faut faire coexister en mémoire centrale la page Ø d'exécution et le moniteur de pagination. Deux solutions étaient possibles pour le moniteur : ou bien son intégration au superviseur, ce qui suppose l'inamovibilité et allonge le superviseur ou bien son existence en concurrence avec la page Ø dans la mémoire banalisée allouée à l'utilisateur. Cette dernière solution n'alourdit pas le système et la relative indépendance du superviseur et du moniteur facilite la mise au point et les améliorations de chacun d'eux. Si le moniteur est, de plus, conçu réentrant, il peut être partagé par plusieurs usagers de la pagination (ce qui le rapproche du superviseur) et fait profiter le système entier de l'allègement de charge que constitue l'effacement, sans recopie en mémoire virtuelle des segments protégés en écriture dès qu'une telle copie existe, quand de la place est nécessaire en mémoire réelle.

Si le moniteur est réentrant, il lui faut 2 zones, une haute et une basse, le registre de zone haute a toutefois été déjà réservé à la limitation de la page courante et le superviseur ne peut attribuer à un usager que 2 segments à un instant donné le bas obligatoire et le haut facultatif.

a) - restrictions dues au superviseur

Le superviseur ne tolère pas certaines manipulations dans les zones hautes : il est interdit d'y établir des zones tampons d'entrée ou sortie et de façon générale, d'y faire des entrées-sorties (même en mode direct). (Cette restriction permet de limiter les contrôles d'adresse multiples, préalables aux instructions réelles de transfert, et également de déplacer des segments hauts multiplement utilisés sans avoir à rechercher si des transferts prioritaires y sont en cours d'exécution).

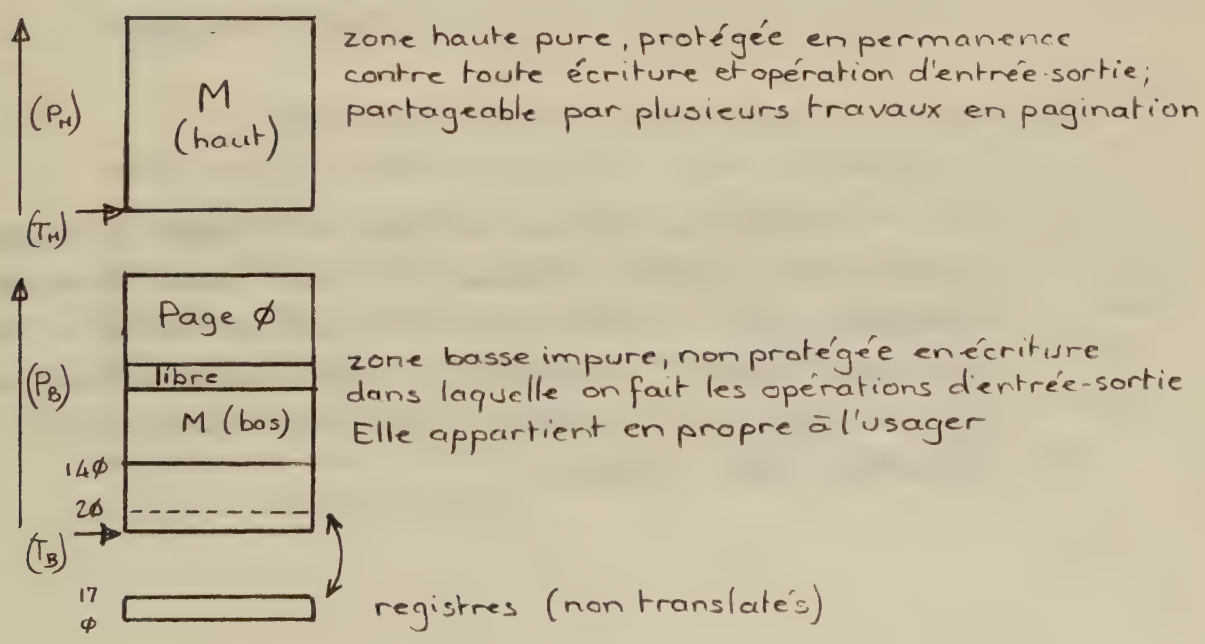
b) - <u>niveau usager de page</u>

Il aurait été très difficile de remettre en cause ce principe et il a mieux valu imaginer un nouveau mode de travail : le niveau usager de page inférieur au niveau usager normal lui-même inférieur au niveau superviseur.

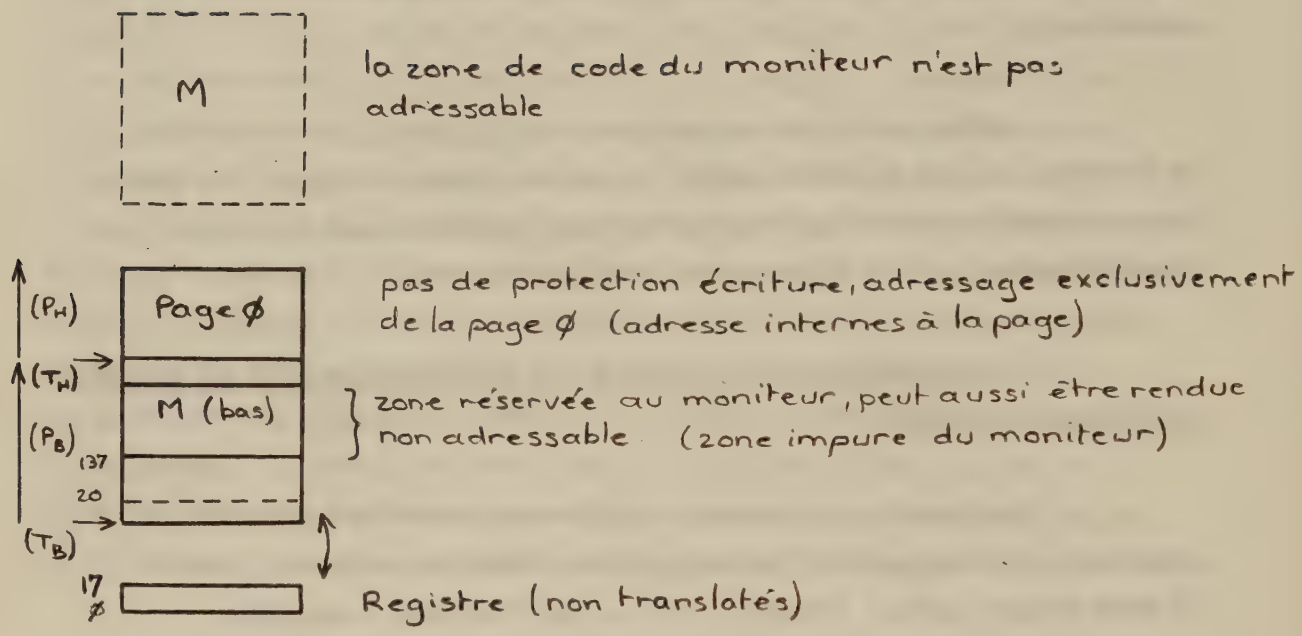
. Selon que c'est le moniteur ou la page Ø qui est en cours de travail, on est en mode usager normal ou usager de page, les translations appliquées et les interprétations possibles aux opérateurs non câblés données par le superviseur sont différentes.

. <u>Position du moniteur vis à vis du superviseur et du système vu dans son ensemble</u>

Programme à 2 segments dont le haut peut être partagé par plusieurs utilisateurs de la pagination. Cette situation est celle du mode usager normal lorsque c'est le moniteur qui travaille.



. <u>Position de la page vis à vis du moniteur</u>



Moyennant une certaine confiance en l'usager de la page, on peut laisser possible l'adressage de la zone impure du moniteur, le préjudice en cas d'altération de cette zone, ne sera qu'à l'égard de son propre travail, la pagination ne permet pas de tourner les règles de sécurité et d'indépendance des divers travaux présents simultanément dans le système.

Le matériel fait que les translations et protections sont inopérantes pour le superviseur, ce ne sont que des outils prêtés aux utilisateurs. En échange de la précaution indispensable de non débordement de sa zone, un usager peut être rendu libre d'utiliser ses registres à sa guise à l'intérieur des deux zones qui lui sont, du point de vue logique, réservées par le superviseur chargé du partage de la ressource "mémoire centrale". C'est cette idée qui a été utilisée pour inclure dans le superviseur cette manipulation des registres de base et de protection selon le mode "normal" ou "page" du travail en cours.

c) - <u>changement de mode</u>

Pour simplifier, un nouvel opérateur programmé a été créé pour que le moniteur demande au superviseur un changement de mode et surtout un changement des registres de base accessibles uniquement en mode superviseur (instructions privilégiées).

Un opérateur programmé commute les 2 registres de zone haute vers le second K de la zone basse et modifie les protections en conséquence, laissant le travail en mode usager-page, la situation inverse est rétablie au prochain opérateur programmé.

Compte tenu du partage de temps qui est contrôlé par le superviseur à un niveau supérieur, il est nécessaire qu'il y ait continuité dans le mode de l'usager : interrompu en mode usager de page, il doit continuer, quand revient son tour d'utiliser l'unité centrale, dans le même mode. Essentiellement les registres de translation et protection doivent être rechargés pour adresser toujours la même zone de programme même si depuis son arrêt l'implantation du programme réel en mémoire a changé ou même cessé (lors d'un séjour en mémoire virtuelle).

Le mode doit être mémorisé comme un état secondaire possible de l'usager, caractéristique de sa zone haute et du renvoi des opérateurs programmés, ce mode sert au moment où le superviseur réinitialise ses registres de translation et protection. Il a suffit d'implanter cette nouvelle possibilité dans le superviseur pour rendre les modes usager de page et usager normal inaltérés par le partage des ressources de base, temps et espace, en mémoire rapide.

d) - modification du superviseur

Les modifications apportées au superviseur ont, en résumé, pour effet :

I°/ - de permettre la commutation des modes usager normal et usager de page , ces modes étant permanents à travers le partage de temps. Ceci est fait par 2 opérateurs programmés ajoutant 30 instructions au superviseur, très faciles à implanter,

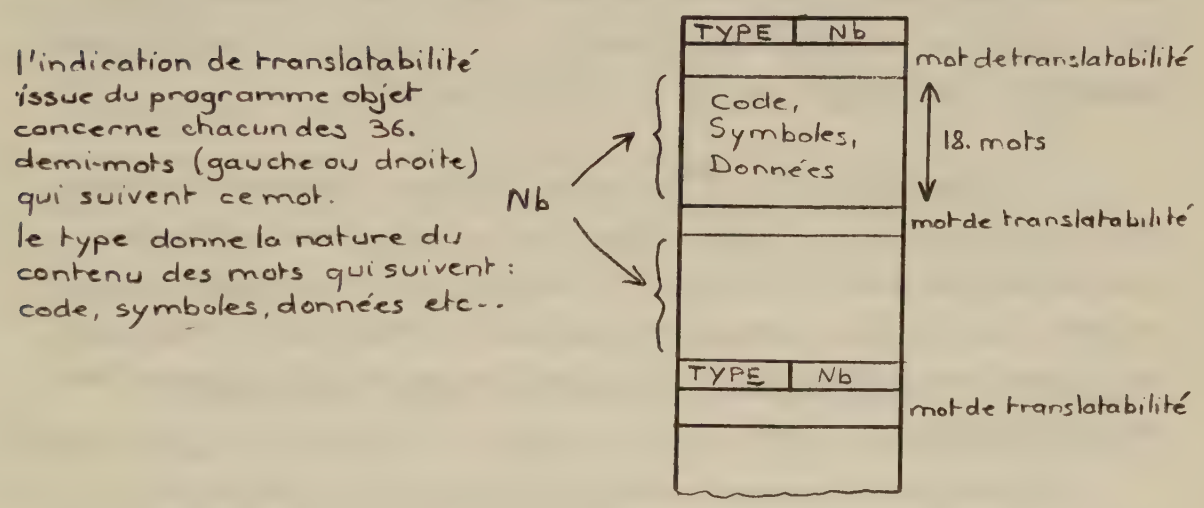
2°/ - de permettre en mode usager de page le renvoi de tous les opérateurs programmés et instructions inexistantes à une adresse précise de la zone de l'usager. Cet état cesse avec le retour en mode usager normal au premier renvoi provoqué.

Le moniteur qui a pris le contrôle lors d'une violation de mémoire pour forcer le changement de mode tentera d'effectuer un opérateur programmé quelconque auquel il se gardera de donner une suite mais qui lui aura permis d'adresser de nouveau sa partie réentrante en étant en mode usager-normal.

7 - ECRITURE DES PROGRAMMES PAGINES

L'idée de départ était que leur écriture ne devait en rien différer de l'écriture de programmes non paginés de manière à pouvoir profiter des bibliothèques de programmes déjà écrits et libérer d'autant le programmeur. Ceci provenait essentiellement de la nature translatable des résultats fournis par les programmes assembleurs et compilateurs habituels sous forme de fichier binaire compréhensible par le programme chargeur de programmes et sous programmes assurant la bonne référence des adresses symboliques globales.

Ce format binaire est le suivant :



Le chargeur interprète, en fonction du type, les données qui suivent cette indication, la majeure partie est constituée de code instructions et constantes qui figureront dans le programme exécutable. D'autres sont les adresses d'implantation de ce qui suit (translatables en général) et les symboles globaux valables pour les autres parties de programme issues de compilations ou d'assemblages différents, avec leur valeur ou un pointeur sur la chaîne de reprise.

En fonctionnement normal du chargeur, les sous-programmes sont chargés en séquence au fur et à mesure , qu'en conversationnel, l'utilisateur du chargeur, à sa console, frappe les noms des fichiers binaires résultants de compilations ou d'assemblages. Pour chacun des sous-programmes aux demi-mots translatables (adresses) (situés à gauche ou plus souvent à droite des mots de code ou de données) le seuil de chargement de ce sous-programme constitué en général de 140 (zone basse) ou de 400000 (zone haute) + la longueur de tous les sous-programmes déjà chargés dans cette zone est ajouté.

En pagination, il s'agit pour faciliter l'exécution de modifier dans chaque page, chaque adresse (une adresse étant un demi-mot translatable de code) en fonction de sa position soit relative au début de cette page si elle indique une position de mémoire de la même page que le demi-mot en question, soit laissée en absolu s'il s'agit d'une page différente.

Ceci est réalisé au chargement et à la translation de chaque demi-mot de code quand on sait encore (le mot de translatabilité n'étant pas conservé par la suite) ce qui est translatable et ce qui ne l'est pas et que l'on connaît l'adresse d'implantation dans le futur programme complet.

La détermination de la nature relative ou absolue de l'adresse après translation est simple puisque la longueur d'une page est également connue. La longueur de la page n'a pas à être fixée à l'écriture

ou à la compilation mais simplement au chargement du programme en précisant au chargeur que l'on veut paginer en pages d'un certain nombre de K.

Ce nouveau mode de fonctionnement du chargeur n'exclut pas, bien sûr, l'utilisation normale, les diverses routines nécessaires à la pagination ont été regroupées en un sous-programme réentrant, la partie réentrante normale du chargeur ayant été modifiée le moins possible.

La transformation en relatif des adresses de page courante, compte tenu de l'inexistence de la page Ø (adresses de 4ØØØØØ à 377777 + longueur de la page), n'a pas entravé la satisfaction des références aux symboles externes. Ces références sont liées entre elles par des chaînes de reprise (pour les demi-mots de droite), chaque cellule pointant sur la référence précédente, ces références sont des adresses non particularisées tant qu'on ne remonte pas la chaîne et comme telles translatables et mises en relatif quand elles pointent en page courante. L'algorithme de reprise doit tenir compte de cela et il n'y a aucune ambiguïté, les pointeurs relatifs étant inférieurs à 377777 + longueur de la page autrement dit pointant la page Ø inutilisée et non chargée.

Le chargeur, après le chargement de tous les sous-programmes précisés dans sa commande, la consultation des bibliothèques éventuelles, la vérification d'unicité des symboles globaux (compatibilité entre programmes écrits indépendamment) et le signalement à l'usager des noms des symboles globaux éventuellement non définis, sauve en mémoire lente sous forme de fichier le segment chargé prêt à l'exécution. Il sauve d'abord un bloc (128.mots) contenant des indications générales non exécutables concernant le programme, l'adresse de départ, la première adresse libre, l'adresse de départ du programme de mise au point s'il existe, la taille d'une page. Ces indications sont en général destinées au superviseur et placées entre les adresses

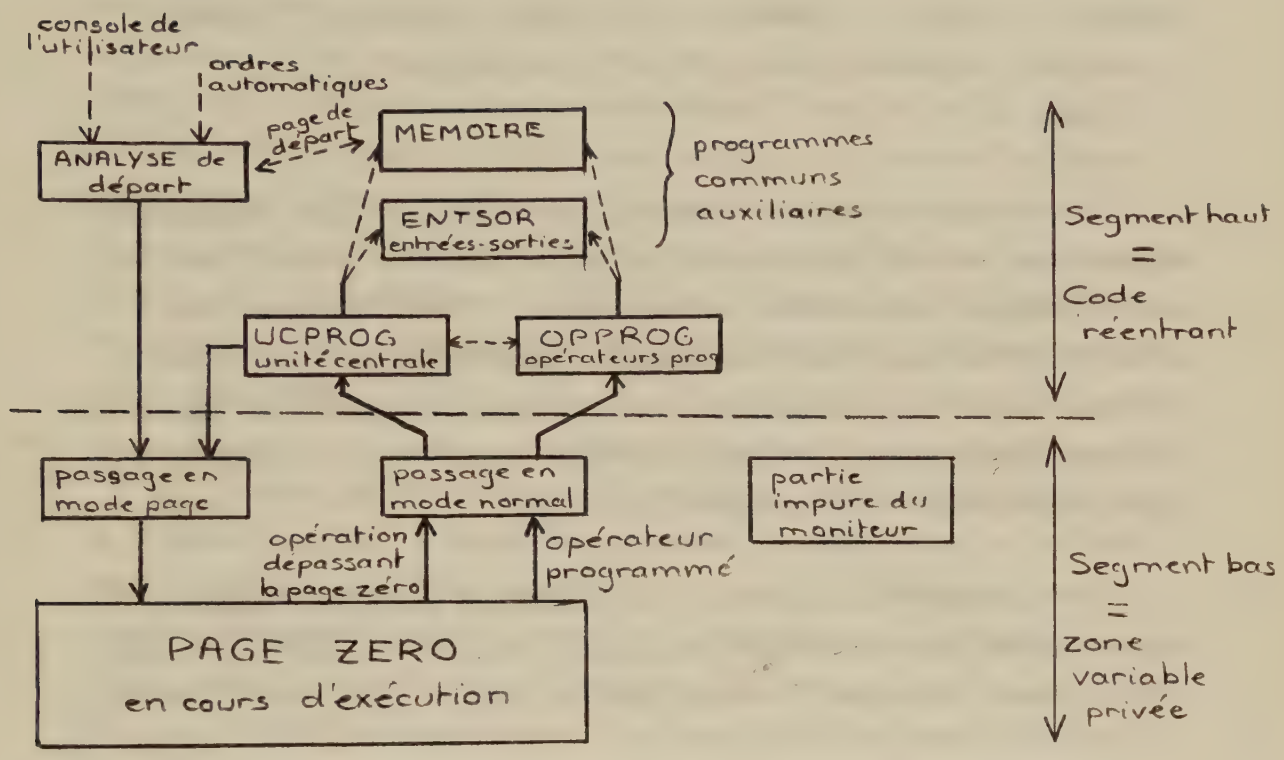
2Ø et 137, elles sont ici destinées au moniteur pour être interprétées au préalable, si elles doivent être passées au superviseur.

Le chargeur sauve ensuite les pages 1 à la dernière chargée (qui contient la fin de la table de définition des symboles au moins globaux) à raison de 8. blocs par K mots de programme.

Le fichier est fermé et le chargeur cesse de fonctionner plaçant ou bien la console de l'utilisateur sous le contrôle du superviseur, ou bien faisant appel directement à un nouveau programme qui le remplace automatiquement, ici, le moniteur.

Le moniteur, s'il est appelé par l'utilisateur attend que celui-ci spécifie le nom du fichier contenant le segment paginé, s'il est appelé par le chargeur, il retrouve le nom à un endroit convenu. Ceci permet d'automatiser les exécutions, l'appel au chargeur pagineur ayant pu être lui aussi le résultat d'une commande automatique, aussi complexe soit-elle, écrite une fois pour toutes et rangée elle-même sous un nom de fichier : ainsi, à l'image de l'adressage indirect, des programmes bien conçus acceptent des commandes indirectes pour lesquelles on ne frappe à la console que l'"adresse",ici, un nom de fichier.

8 - ORGANISATION DU MONITEUR DE PAGINATION - ANALYSE DE DEPART



L'analyse de départ consiste en l'attente d'une ligne de commande tapée à la console de l'utilisateur ou bien d'une commande stockée en un lieu convenu (ceci n'a pas été mis au point). Le texte est analysé pour retrouver essentiellement le nom du fichier contenant le programme paginé à exécuter. La commande générale est :

ORGANE :	FICHIER .	QUALIFICATIF	[REPERTOIRE]	28
facultatif valeur implicite = DISQUE:		facultatif valeur implicite = .SOV	facultatif sur le disque, zone ou réper- toire contenant le fichier	facultatif limite en K. à l'expansion de mémoire

L'usage du disque comme support de fichier est pratiquement obligatoire, mais il est bon que le nom puisse être précisé pour le cas des disques particuliers privés ou non, DISQUE: ne constituant qu'un générique pour toute une classe de périphériques similaires. Une bande magnétique adressable pourrait aussi être utilisée moyennant une modification simple du module MEMOIRE chargé de la recherche des pages et fragments de pages dans le fichier.

L'intérêt du système va en diminuant quand des supports à temps d'accès plus longs sont utilisés. L'analyse conduit à l'ouverture du canal programmé Ø initialisant le disque logiquement concerné.

Le fichier dont le nom a été trouvé dans la commande est ouvert en lecture et en écriture dans le but d'être mis à jour en permanence avec les pages qui ne sont plus utilisées et qui ont peut-être été modifiées, il n'existera qu'une seule version du fichier où l'on pourra relire ce qui fut écrit peu de temps avant. Ces ouvertures sont suivies de la lecture du premier bloc du fichier contenant des renseignements essentiels tels que la longueur d'une page et l'adresse absolue de départ. La fin de ce bloc contient en outre des pointeurs naturellement absolus, vers certaines cellules du programme dont la partie droite mérite d'être mise en absolu, car elle risque d'être transmise d'une page dans une autre sans prendre garde qu'elle était relative à sa page d'origine et sans valeur hors de cette page. C'est un des risques de la pagination de trans-

mettre d'une page à une autre des pointeurs ou des adresses relatives (par exemple, les pointeurs de pile conservés dans les registres, communs à toutes les pages,doivent être dès le départ mis en absolu). Cette méthode est la plus simple qui ait été trouvée sans modifier les assembleurs ou compilateurs déjà existants. Il suffit ici de créer et de faire assembler un petit programme assembleur (non exécutable car ne contenant que des pointeurs) localisé de 152 jusqu'à 177, pointant à chaque mot vers une position (globalement déclarée) du programme paginé, le chargeur charge ce code non translatable dans le premier bloc (sorte de vestige de la page Ø) en valeur absolue. Ce code est interprété par ANALYSE avant l'exécution pour compenser les mises en relatif excessives et que l'on ne peut inhiber, faites par le chargeur.

Après ces opérations, l'adresse de départ est chargée dans le pointeur servant de compteur programme ; avant de passer en mode page, on vérifie systématiquement que la page courante contient le compteur programme absolu (et dans le cas contraire, on change de page), après quoi ce compteur programme est mis en relatif, le mode page est demandé au superviseur. Les registres (ici remis à Ø) sont rendus à l'utilisateur qui prend le contrôle en page Ø. Cette dernière partie de code est écrite en zone basse car dès la mise en mode usager de page, le segment réentrant du moniteur n'est plus accessible, la translation haute pointant dorénavant sur le bas de la page zéro.

Si on envisageait de ne pas permettre l'adressage de la partie impure du moniteur en mode usager-page, il faudrait que les dernières fonctions soient effectuées par le superviseur lui-même ; cette restriction n'a pas été envisagée car son intérêt est limité.

Le mode usager-page se conserve jusqu'à la prochaine cause d'interruption venant de la page zéro elle-même.

9 - INTERRUPTIONS POUR DEPASSEMENT DE LA PROTECTION

Dans le superviseur, au niveau prioritaire attribué à l'unité centrale, la routine chargée des cas de dépassement de la protection de mémoire remarque que l'utilisateur courant a demandé que ce cas d'interruption soit signalé à son programme et non pas tenu pour une erreur fatale arrêtant son travail en cours. Le moniteur fait une telle demande avant tout relancement du mode usager-page. Dans ces conditions, la violation de mémoire consécutive au mode d'adressage en absolu de toute quantité extérieure à la page courante d'un programme paginé ne manquera pas d'être signalée au moniteur, aussitôt détectée. La routine d'interruption du superviseur range alors le compteur programme dans une position connue de la zone de l'usager (entre 2Ø et 137) (JOBTPC) ainsi que l'état de l'unité centrale à ce moment. L'adresse de traitement d'interruption de l'usager (ici le moniteur) est placée dans le compteur programme de la machine avant de faire cesser l'interruption réelle ; elle est indiquée en JOBAPR (entre 2Ø et 137). Le branchement est ainsi décidé autoritairement par le superviseur du programme en mode usager de page vers le moniteur chargé d'interpréter l'arrêt et d'y remédier.

Pour l'instant, seule la zone basse du moniteur est adressable, une des premières opérations est de repasser en mode usager normal au moyen d'un opérateur programmé factice non interprété.
Le moniteur sauve les registres du programme paginé et commence la recherche de la cause de l'interruption grâce au mot donnant l'état de l'unité centrale quand la violation de mémoire a été tentée.
Le cas d'une interruption pour une autre raison qu'un dépassement de la protection mémoire n'est possible que si le programme paginé a demandé au superviseur à travers le moniteur, d'en effectuer le traitement lui-même, le moniteur simule, dans ce cas, l'interruption à l'égard du programme paginé comme le superviseur le fit à son égard.

a) - Les différents cas

A/ - Un cas simple causant une interruption est le saut hors de la page Ø à une adresse absolue ou une adresse provenant de l'évolution naturelle (+ 1 ou + 2) d'une des toutes dernières adresses de la page en cours venant de page Ø en page 1 mais on se doute que le saut est fait à la page qui suit la page courante et dont on calcule l'adresse absolue. Après réécriture en bonne place sur la mémoire lente (fichier) de la page courante abandonnée puis accès à la page demandée, le compteur programme de reprise est mis en relatif pour la nouvelle page zéro, le mode usager de page est rétabli et le contrôle repassé à la nouvelle page zéro comme au départ. Ce cas, où la recherche de l'instruction a provoqué la violation de mémoire ne présente pas d'autre difficulté que l'adressage absolu des 2 premières positions de la page n° 1 en tant que positions de branchement : il est bon de ne pas les utiliser pour s'y brancher pour éviter toute ambiguïté.

B/ - Dépassement de protection pour une autre raison que le saut à l'extérieur de la page en cours

On se heurte rapidement à la difficulté que constitue l'ambiguïté. L'ambiguïté est due au mécanisme de protection mémoire et à l'action entreprise en cas de dépassement.

Les accès-mémoire de l'unité centrale se font (cf. annexe A) :

a/ - à l'accès de l'instruction dont l'adresse est prise dans CP ou implicite (branchement autoritaire vers une "trappe") et placée dans AM (voir plus haut).

b/ - au calcul d'adresse, si au cours de la recherche d'adresse, après indexation éventuelle et indirection, il y a violation de la protection, l'instruction est exécutée quand même avec une adresse nulle (ceci peut détruire le contenu du registre Ø de façon indésirable, aucune réparation n'est possible), ceci condamne tout saut indirect en dehors de la page courante.

c/ - à la recherche des opérandes, une adresse trop grande provoque l'arrêt immédiat de l'exécution, le compteur programme CP pointe toujours sur l'instruction coupable.

d/ - il n'y a pas d'accès mémoire au cours de la phase d'exécution, les instructions multiples repassent par la recherche d'opérande et même le calcul d'adresse (pointeur de champ).

e/ - au rangement des résultats, le compteur programme est incrémenté, certaines instructions se singularisent par un rangement de l'ancien compteur programme après mise en place d'un nouveau issu du code d'instruction. Pour ces instructions utilisées ou susceptibles d'être utilisées en mode multipage, la meilleure solution est le remplacement par un opérateur programmé strictement équivalent mais adapté à tous les cas ; il faut en particulier, pour que la reprise ultérieure (appel à sous-programme) ne soit pas ambiguë, sauver un compteur programme "absolu" ce que ne sait pas faire le matériel qui manipule les adresses de la page Ø. Les 4 instructions de piles ont intérêt à être remplacées par des opérateurs programmés (instructions inexistantes de codes 1ØØ, 1Ø2, 1Ø3, 1Ø4).

b) - Ambiguïtés rencontrées et solutions proposées

Il apparaît indispensable de maîtriser l'exécution de

appel : PUSHJ REGISTRE,ADRESSE en usage multipage ou avec une pile pointée par registre, toujours inacessible dans ce dernier cas, le retour avec descente de pile

retour : POPJ REGISTRE,Ø est sans ambiguïté si le compteur programme sauvé l'a été en absolu (retour à la bonne page et non à la dernière page utilisée)

appel : JSA REGISTRE,ADRESSE en usage multipage doit sauver le contenu de REGISTRE en ADRESSE, passer en ADRESSE + 1 en conservant ADRESSE et le compteur programme (absolu) au moment de l'appel + 1

retour : JRA REGISTRE,ADRESSE n'est pas effectuée si ADRESSE au moment du JSA d'appel a été mis en absolu (arrêt à la recherche de l'opérande)

JSR ADRESSE en usage multipage, il y a ambiguïté sur la page de retour, il faut assurer la mise en absolu au moment de l'exécution

JSP REGISTRE,ADRESSE en usage multipage soulève une ambiguïté au retour, cet opérateur doit être simulé par un opérateur programmé pris parmi les instructions inexistantes.

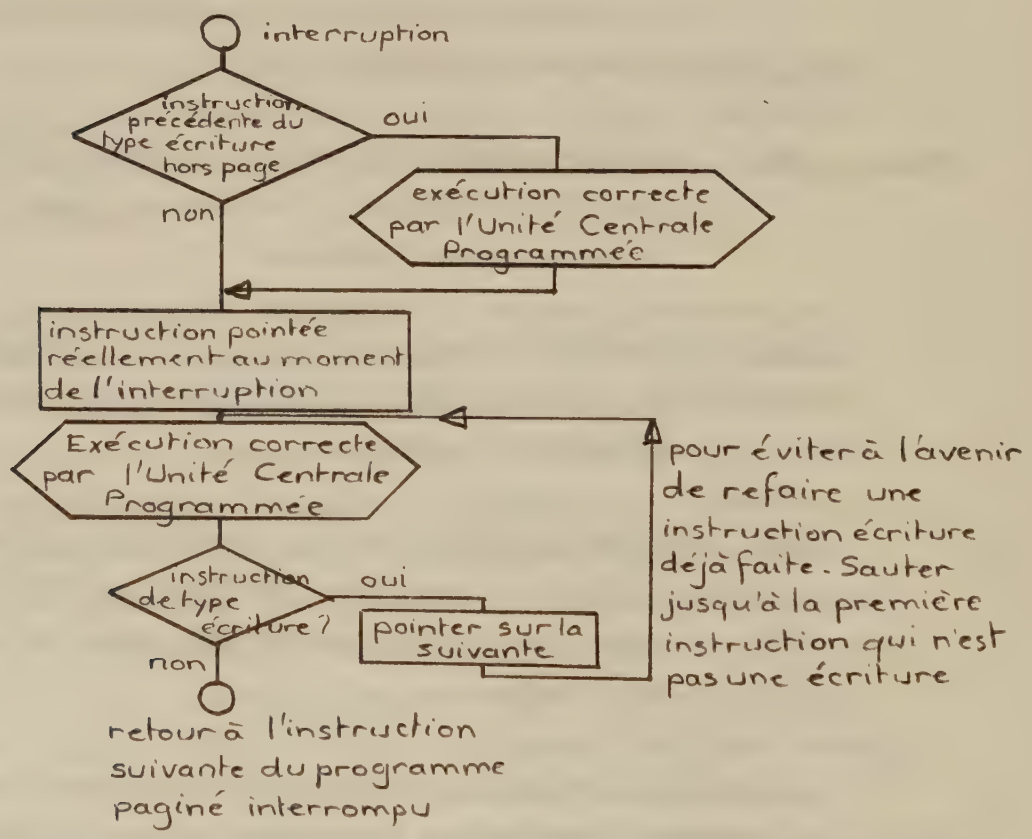
De façon générale ce sont les instructions avec écriture mémoire non précédée d'une lecture de la même mémoire qui apportent des ambiguïtés. L'exécution programmée de PUSHJ, JSA, JSR, JSP permet de ne jamais perdre d'information en usage multipage. Une façon simple

de faire le changement de code est de redéfinir en tête du fichier à assembler la signification binaire des opérateurs symboliques, la définition étant prioritaire sur la valeur implicite que donnerait l'assembleur.

Il demeure malgré tout une ambiguïté lorsqu'un compteur programme trouvé relatif à la page en cours peut aussi bien pointer

a) - sur l'instruction fautive qui doit effectivement présenter une violation en accès aux opérandes,

b) - sur l'instruction suivant l'instruction fautive qui ne doit pas présenter de violation en accès aux opérandes, mais doit en présenter au rangement des résultats, d'où l'ordre de recherche de l'instruction fautive :



La recherche (et l'exécution préalable en cas d'apparence de faute = écriture hors de la page) de l'instruction qui précède celle pointée peut entraîner des ennuis car ne pas correspondre à l'exécution réelle du programme. Pour lever ces ambiguïtés, il est bon d'intercaler une non-opération entre une instruction du type écriture (arrêt avec CP incrémenté), adressant (peut-être) une position absolue (pas en page Ø) et une instruction du type lecture (arrêt avec C.P. non incrémenté) adressant (peut-être) une position absolue du type lecture (arrêt avec CP non incrémenté) ou ce qui est ici équivalent, du type lecture, puis écriture. Ce cas est automatiquement signalé pour toutes les instructions étiquetées, par le chargeur fonctionnant en mode paginant puisque les instructions étiquetées devraient être les seules auxquelles un branchement est exécuté.

Dans le cas d'un branchement possible à l'étiquette, il serait malvenu d'effectuer l'instruction précédant l'étiquette sous prétexte qu'elle est susceptible d'avoir provoqué l'interruption pour dépassement de protection, une non-opération peut par exemple lorsque l'ambiguïté a été signalée au chargement et vérifiée par le programmeur sur la liste du sous-programme chargé, être introduite à l'étiquette pour éliminer cette possibilité d'ambiguïté. Le moniteur ne signale pas les ambiguïtés, il obéit simplement à l'algorithme indiqué ci-dessus, il a beaucoup moins d'éléments que le chargeur pour supposer une ambiguïté (absence des étiquettes) tout en ayant l'avantage de pouvoir simuler l'exécution pour voir si elle est possible sans dépassement de la protection (disparition du peut-être).

Le traitement de l'interruption provoquée par un besoin d' argument situé en dehors de la page courante, est effectué par l'unité centrale programmée UCPROG qui exécute les instructions sur des arguments retrouvés, au moyen du module MEMOIRE qui, lui, contient les algorithmes d'interface et de changement de mode d'adressage (mode absolu et mode relatif à la page en cours).

Le module UCPROG contient un tableau autogénéré par macro-assemblage permettant d'identifier pour un code opération donné, le type de l'instruction : par exemple, instruction de type lecture ou lecture puis écriture, instruction de type écriture, instruction de rupture de séquence, instructions particulières.

En fonction du code de genre d'instruction, l'exécution se fera par montée de pile à une sous routine spécifique chargée de préparer les arguments ou de les ranger, l'exécution étant soit simulée, soit réelle par une instruction XCT (exécuter) pour les codes simples.

Moins d'une vingtaine de genres d'instructions a pu être isolée parmi les 365 instructions câblées possibles, ou plutôt les 384 codes possibles avec les instructions inexistantes servant d'opérateurs programmés de remplacement.

Dans UCPROG une routine unique se charge des calculs d'adresse, des indexations et de l'indirection indéfinie en veillant à remettre en absolu chaque adresse en fonction de sa page d'origine.

D'une façon générale, le moniteur s'efforce de transformer pour ses propres usages, les adresses relatives en adresses absolues, pour éviter toute ambiguïté, en particulier au moment des accès à la mémoire.

La simulation des instructions de pile tient compte de la possibilité de dépassement de capacité de la pile, si l'utilisateur en a fait la demande au superviseur, ce cas est signalé à son programme par un branchement à la même adresse que celle de l'unité centrale quand ce n'est pas une violation de mémoire.

Au module mémoire, on demande au moyen d'une adresse absolue, soit un mot de la page courante ou d'ailleurs,
soit une page, celle qui contient l'adresse indiquée devant remplacer la page courante.

Pour limiter les accès au fichier contenant toutes les pages du programme, pour la même adresse ou pour des adresses voisines, compte tenu du mode réel d'adressage de la mémoire lente (bloc de 128 mots au minimum), le dernier fragment lu de cette mémoire est conservé en mémoire rapide dans la zone basse du moniteur. Ce fragment ne doit être remplacé qu'en cas de besoin d'un nouveau fragment et ne doit être récrit en mémoire lente que s'il a été modifié. La modification est signalée par les routines qui ont modifié le fragment et qui le savent, simplement la recopie en mémoire lente n'est pas faite au moment où elle est demandée, un indicateur précise qu'avant remplacement, ce fragment devra être recopié en mémoire lente (fichier) dans la bonne page.

Il faut veiller quand on charge une nouvelle page qu'une version plus récente d'un de ses fragments peut déjà être en mémoire rapide, le fragment de page doit alors être recopié par déplacement en mémoire centrale dans la nouvelle page entrée, la version à jour devient la page et non plus le fragment.

Néanmoins, un fragment peut très bien séjourner plusieurs interruptions d'affilée, s'il est seul à être référencé à chaque interruption (cas d'une boucle ou d'accès successifs à des positions de mémoire voisines situées dans le même fragment).

Un système plus complexe pourrait laisser séjourner plusieurs fragments de pages couramment utilisés pour limiter encore davantage les accès à la mémoire lente, les contrôles seraient plus longs, mais toujours beaucoup plus rapides qu'un transfert mémoire lente - mémoire rapide, à condition bien sûr que de la place soit disponible pour le stockage de ces fragments sans allongement de la zone basse du moniteur (adresses 14Ø à 1777 en pratique).

Le module MEMOIRE est également chargé de la simulation de l'opérateur programmé de gestion de mémoire lorsque l'attribution de nouveaux K de mémoire est demandée, jusqu'à concurrence de l'argument facultatif fourni au lancement du programme paginé ou à défaut, jusqu'à 128. K ou jusqu'au remplissage du support du fichier. Il ne fait les expansions que par pages entières pour inclure la plus haute adresse (absolue) désirée, passée en argument.

La gestion des fichiers ne permet pas, actuellement, une réduction de la taille d'un fichier. L'allongement seul est prévu par l'écriture de blocs emplis de zéros.

10 - INTERRUPTIONS PAR LES OPERATEURS PROGRAMMES OU LES INSTRUCTIONS ILLEGALES

Après l'aménagement apporté au superviseur pour le cas du mode usager de page, les interruptions par opérateur programmé ou instruction illégale (non câblées ou interdite en mode usager) provoquent le passage automatique en mode usager normal et le branchement à l'adresse 41 du programme, adresse menant au moniteur lui-même, l'adresse relative incrémentée, de l'instruction responsable est mémorisée en UUOØ et placée en absolu en PCPAGE avant traitement de l'opérateur programmé et après sauvegarde des registres. En UUOINT les opérateurs programmés propres à l'usager de page lui sont renvoyés s'il dispose d'une adresse de renvoi en simulant l'interruption comme le fait à l'égard du moniteur, le superviseur lui-même.

Pour certains opérateurs du programme contrôleur des programmes Fortran, certaines mises en absolu sont souhaitables à ce niveau pour éviter des ambiguïtés dues à des changements de page d'arguments passés par les registres.

Les instructions illégales sont filtrées et simulées par l'unité centrale programmée s'il s'agit de codes utilisés pour remplacer les opérations câblées mal exécutables en cas de dépassement de protection (voir plus haut).

Les autres opérateurs programmés sont identifiés au moyen de tables d'adresses de routines qui en feront l'exécution spécifique selon le code et les précautions à prendre dans chaque cas avant le passage de l'opérateur programmé au superviseur (souvent par exécution au moyen d'une instruction XCT).

Pour les codes d'instruction 40 et 47 qui sont en fait des appels à des sous-programmes du superviseur, il est fait usage d'une seconde table donnant le type de l'opérateur et l'éclatement est fait après consultation d'une troisième table de type.

Ces tables sont définies par une macroinstruction unique pour en faciliter l'expansion future. Les opérateurs programmés les plus simples sont passés tels quels au superviseur après restitution momentanée des registres de l'utilisateur de page (routine XCTUUO). Toutes les routines sont prévues pour être appelées par une montée de pile où l'adresse est sauvée et se terminent par retour à l'adresse pointée par la pile et descente de pile. L'adresse de retour peut dans certains cas avoir été incrémentée au cours du passage dans les programmes appelés, pour rendre possible d'éventuels retours multiples.

Les prétraitements spécifiques des manipulations de données, de tampons, de fichiers et d'équipements ont été regroupés dans le module ENTSOR. Ce module doit tenir compte :

1°/ - de l'occupation définitive du numéro logique d'équipement zéro comme canal d'échange programmé avec le fichier de mémoire lente portant les pages et réservé au moniteur. Pour y parvenir il faut tenir à jour une table (4 bits par numéro logique) des numéros de canaux programmés déjà utilisés avec attribution du premier libre en cas de non inscription et effacement lors de l'abandon de l'équipement. Ainsi les programmes des utilisateurs peuvent faire un usage quelconque des numéros logiques d'équipement (Ø à 15.) placés dans chaque opérateur programmé d'entrée-sortie à la place normale du registre.

CNVCAN assure la conversion de numéro de l'utilisateur des pages en numéro connu du superviseur qui connaît déjà le Ø comme fichier support des pages du programme.

2°/ - des différences dans le mode d'adressage de certaines quantités communes à l'usager de page (adresse absolue ou relative) et au superviseur (adresse locale du segment bas du moniteur) qui contient la page Ø en PAGLOC = 2ØØØ. Ceci oblige à des modifications d'absolu en local des pointeurs de tampon courant et à la

vérification que tous les tampons d'un même canal d'échange programmé sont dans la même page et les échanges par tampons sont terminés avant de repasser le contrôle au mode usager de page. On profite de l'avantage du travail par bouffées de tampons sans la simultanéité qui est incompatible avec le risque de changement de page courante pour laquelle un transfert est déjà en cours, car le système n'interdit pas de sortir une zone mémoire sur un périphérique, alors qu'une écriture y est en cours venant d'un autre périphérique (l'interblocage ultime est sur l'accès au mot de la ferrite par l'unité centrale et le canal direct sur 2 portes différentes ou sans canal, par les interruptions d'origines différentes de l'unité centrale). Un tel cas s'il est possible, n'est pas tolérable et encore moins 2 écritures (ou plus), venant de 2 périphériques différents pour la même zone mémoire. Un opérateur programmé permet l'attente (au profit d'un autre travail) de la cessation d'activité d'un équipement reconnu par son numéro de canal programmé, il n'a d'utilité que lors de l'usage de tampons multiples pour des transferts.

Le moniteur, pour les opérateurs programmés d'entrée-sortie, doit effectuer les contrôles et modifications nécessaires pour, dans chaque cas, éviter un diagnostic fatal et désagréable du superviseur. Malgré tout, il doit permettre les cas de figures les plus courants avec le minimum de restriction. Certaines restrictions sont apparues qui sont :

- la nécessité d'avoir tous les tampons d'un même canal programmé dans la même page pour être accessibles d'un seul coup par le superviseur,

- la décomposition des ordres de transfert direct (sans tampon) en multiples entiers de l'enregistrement normalisé dans le cas de chevauchement de page, au total l'ordre est exécuté, le résultat peut différer par des espaces inter-enregistrements plus nombreux dans une sortie sur bande magnétique. Dans le cas de l'entrée, il n'est pas possible de lire un enregistrement plus long qu'une page (sur bande uniquement).

- tous les changements de page et les contrôles alourdissent beaucoup les opérations d'entrée-sortie, leur volume a intérêt à être limité en pagination ou bien à être judicieux avec un programme écrit spécialement pour ce cas. De façon générale, on s'est efforcé de rendre paginable tout programme écrit de façon convenable comme le font les compilateurs mais des résultats meilleurs peuvent être obtenus en écrivant en tenant compte des futures pages.

  Cette écriture meilleure paraît difficile au niveau des compilateurs déjà en place mais pourrait être rendue optionnelle pour le chargeur si on lui ajoutait des règles d'initiative dans l'implantation en page unique des sous-programmes, ceci allongerait les segments en perdant de la place avec des pages non pleines mais limiterait les changements de page en exécution (cas d'une boucle à cheval sur deux pages).

11 - CONCLUSIONS SUR CE SYSTEME DE PAGINATION PROGRAMME

Des modifications pourront toujours être apportées pour améliorer les performances sans nuire à la généralité mais un des objectifs de départ échappera toujours à sa réalisation ; c'est celui de l'utilisation sans remaniement des programmes ou sous-programmes translatables des bibliothèques. Cette difficulté est apparue au fur et à mesure de la réalisation et des essais du moniteur quand des instructions ont mené à des situations irrécupérables par le moniteur et des ambiguïtés ont été constatées sous forme d'erreurs. Les instructions dont l'exécution est dangereuse pour la suite des opérations (faute de leur page d'origine) ont été remplacées par de nouveaux opérateurs programmés en nécessitant un réassemblage (programmes en langage d'assemblage), elles n'étaient heureusement pas générées par le compilateur FORTRAN. Les risques d'ambiguïté ont été mis en évidence pour les instructions étiquetées par le chargeur en mode paginant dans le but de signaler au programmeur les endroits où il doit prévoir des non-opérations qu'il insère en modifiant ses cartes ou son fichier-programme source et qui nécessite un réassemblage. Le compilateur FORTRAN ne paraît pas générer de séquences ambiguës, auquel cas l'action serait difficile sans modifier le compilateur lui-même.

En résumé, le matériel étant celui décrit en annexe, il n'est pas possible de paginer de façon sûre sans un réexamen des programmes source sous peine d'aboutir à des ambiguïtés fatales ou a des erreurs d'exécution. La simulation de l'unité centrale ferait bien sûr disparaître ce problème mais au prix d'un temps d'exécution encore plus considérable alors qu'actuellement, ce temps est déjà multiplié par un facteur compris entre 10. et 100. pour des programmes dont le code couvre plusieurs pages. Sauf les cas marginaux de programmes très gros qui, en conséquence, méritent d'être écrits et implantés de façon à minimiser les accès hors page et les changements de pages, le système de pagination étudié ici pour le système déjà existant, n'apparaît pas concurrentiel face à l'exécution directe normale. Cette étude a

néanmoins contribué indirectement à la mise en évidence de certaines failles du matériel (exécution avec adresse zéro) et surtout du système d'exploitation en temps partagé lui-même. Seul ce dernier a été modifié en conséquence, faisant ainsi disparaître certaines des causes des blocages aléatoires encore inexpliqués.

Des modifications du matériel pourraient être envisagées qui léveraient les ambiguïtés signalées plus haut et faciliteraient la suite à donner aux adressages absolus violant la protection de mémoire.

Ce qui est malgré tout intéressant, c'est la façon d'adresser directement dans une instruction un mot d'un fichier (implicite) à charge pour le moniteur de délivrer ce mot convenablement. Le cas particulier d'une seule page de code et de multiples pages, toujours en mémoire lente, contenant les données à traiter permet de résoudre le problème des grands tableaux, évoqué initialement. Le fichier peut ainsi contenir des pointeurs non ambigus que la page de code utilise avec indexation et indirection au besoin, comme des pointeurs en mémoire rapide.

L'armature du fichier bien qu'elle ne soit pas un programme exécutable peut très bien être générée en langage d'assemblage (symboles, en particulier extérieurs) assemblée et mise en place au moyen du chargeur pagineur réalisé. Ceci est à rapprocher de la technique plus générale de l'adressage symbolique absolu imaginée pour un matériel beaucoup plus adapté et des bibliothèques de programmes spécialement conçues pour ce type d'utilisation (1).

Ce mode d'adressage pourrait trouver des applications utiles tout particulièrement dans les structures de listes ou pour certains fichiers.

(1) - P. DEBRAINE - Adressage symbolique absolu - Structure facilitant le travail en partage de temps - Rapport CEA-R-3588 - 1968

CHAPITRE 5

ANNEXE A

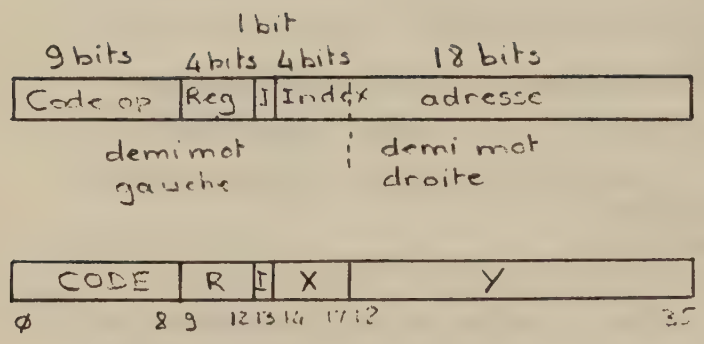
LE MATERIEL UTILISE

I - LES CODES D'INSTRUCTION

1 - Choix d'une taille de mot machine

La plus petite unité d'information retrouvable en mémoire rapide doit être courte par économie de support rapide et de registres
doit être longue pour avoir un code d'instruction riche
un adressage sélectif important
l'idéal serait peut-être la machine à octet ou à mots de longueurs variables avec les complications inhérentes.

Une solution consiste à n'adresser que des mots entiers mais à disposer d'instructions susceptibles de saisir une fraction unique de ce mot : par exemple, sa partie gauche ou droite et au moyen d'un second mot, tout caractère, toute suite de bits du mot adressé au moyen d'autres instructions ; l'usage et la manipulation des pointeurs est facilitée si toute adresse tient précisément dans un tel demi-mot, l'autre pouvant contenir le code instruction, le registre et les conditions d'adressage (indexation, indirection). Comme pour adresser plus de 100 K mots, il faut plus de 16 bits, un demi-mot de 18 bits, un mot de 36 bits répondent en outre à des problèmes de compatibilité de matériel avec d'autres séries de machines. Le format des instructions peut être unique pour faciliter le travail de l'unité centrale, en offrant un large jeu d'instructions.



ceci permet de coder 16. registres généraux de travail Ø à 15. et 15. registres d'index 1. à 15. La zone de codage du registre peut pour certaines instructions faire une extension du code de l'instruction, la zone I, X, Y est toujours décodée pour fournir une quantité de 18 bits $E = Y + (X)_{18-35}$

si $I = Ø$ (modulo 2^{18})

si $I \neq Ø$ $(E)_{18-35}$ remplace I - X - Y jusqu'à ce que $I = Ø$

2 - <u>Classification des instructions</u>

Le code opération sur 9 bits permet 512 instructions de base, il y en a en fait 365 en tout (1970) qui sont exécutées par l'unité centrale

317 instructions de base internes
40 instructions optionnelles (opérations flottantes et sur les caractères)
8 instructions d'entrées-sorties adressant 128 équipements dont 126 quelconques

27 codes sont encore disponibles pour des extensions
33 codes font appel à une adresse fixe unique et non translatable en mode superviseur
31 codes font appel à une adresse fixe translatable sans changement de mode

Deux modes de travail sont retenus pour l'unité centrale,

le mode normal pour lequel les registres de base ne s'appliquent pas et toutes les instructions sont permises,

le mode restrictif ou utilisateur à qui s'applique la translation systématique par registre de base et l'interdiction des instructions d'entrée-sortie.

Seules les interruptions extérieures, l'utilisation d'une instruction interdite ou non encore implantée ou l'un des 33 codes d'appel au mode normal permettent de sortir du mode restrictif pour passer au mode normal ou exécutif.

L'état arithmétique de l'unité centrale est lisible avec le compteur programmé (18 bits) en un seul mot machine,

2 bits indiquent le mode,

6 bits indiquent les dépassements de capacités et les reports arithmétiques.

3 - Revue des instructions de base

a) - transferts de demi-mots

gauche vers gauche	autre moitié inchangée	mémoire vers registre
gauche vers droite	emplie de 1	adresse vers registre
droite vers gauche	emplie de Ø	registre vers mémoire
droite vers droite	extension du signe	mémoire vers même mémoire et vers registre ≠ Ø

L'extension du signe d'un demi-mot transféré au mot entier constitué permet de lui appliquer ensuite les instructions arithmétiques sans ambiguïtés de signe.

b) - transferts de mots entiers

contenu	de mémoire vers registre
valeur négative du contenu	quantité immédiate vers registre
valeur absolue du contenu	de registre vers mémoire
2 moitiés échangées du contenu	mémoire vers même mémoire et vers registre ≠ Ø

c) - transfert multiple, mémoire mémoire, interruptible

Transfert Registre, adresse limite

avec Registre

adresse ler mot source	adresse ler mot destination
Ø 17	18 35

en cas d'interruption, le contenu de Registre est pour la reprise, l'adresse du ler mot source suivant, adresse du premier mot destination suivant.

d) - <u>opérations arithmétiques en virgule fixée</u>

ajouter	registre	0	registre	dans registre
soustraire	registre	0	adresse	dans registre
multiplier				
multiplier comme entier	registre	0	mémoire	dans mémoire
diviser	registre	0	mémoire	dans mémoire
diviser comme entier			et	dans registre $\neq \emptyset$

La représentation est en complément à 2, la virgule est supposée à droite du bit de signe (bit $\emptyset$).

Les opérations entières considèrent la virgule à droite du bit de plus faible poids (35). Ces instructions sont recommandées pour manipuler les faibles quantités : adresses, caractères, indices.

e) - <u>opérations de décision et de transfert de commande</u>

- sans incrémenter / en incrémentant / en décrémentant
  - le contenu de la mémoire et en changeant le registre $\neq \emptyset$
    - sauter l'instruction suivante si le résultat est
      - jamais
      - négatif
      - nul
      - négatif ou nul
      - toujours
      - positif
      - positif ou nul
      - non nul
  - le contenu du registre
    - sauter à l'adresse si le contenu du registre
      - jamais
      - si inférieur
      - si égal
      - si inférieur ou égal
      - toujours
      - si supérieur
      - si supérieur ou égal
      - si inégal

- comparer le contenu du registre
  - au contenu de la mémoire à l'adresse
    - sauter l'instruction suivante
      - jamais
      - si inférieur
      - si égal
      - si inférieur ou égal
      - toujours
      - si supérieur
      - si supérieur ou égal
      - si inégal

Pour l'exploration des tableaux, 2 instructions très utiles permettent un branchement quand celui-ci est épuisé : compte de mots négatif initial en partie gauche du registre (d'index)

- incrémenter chaque moitié du registre
  - sauter à l'adresse si le résultat
    - est positif
    - est négatif

f) - <u>opérations de transferts inconditionnels</u>

Ils permettent de sauver ou non le compteur programme et l'état arithmétique de l'unité centrale, de tester, modifier les drapeaux d'état sous réserve de la hiérarchie des modes, l'arrêt de l'unité centrale (en mode normal seul).

Le sauvetage a lieu en mémoire, dans un registre lui-même sauvegardé automatiquement ou non (imbrication).

g) - <u>opérations câblées pour manipuler les piles</u>

Le pointeur de pile est contenu dans un des 16 registres généraux.

REGISTRE	compteur $\gtrless$ Ø	adresse de haut de pile

montée de pile par incrémentation des 2 moitiés du registre — interruption si dépassement (compteur = Ø) — transfert mémoire vers pile rangement drapeaux,, compteur programme déjà incrémenté sur la pile

transfert pile vers mémoire transfert pile vers registres d'état arithmétique et compteur programme — descente de pile par décrémentation des 2 moitiés du registre — interruption si dépassement (compteur = - 1)

h) - opérations logiques

- les 16 fonctions possibles entre 2 variables binaires sont câblées : combinaison des bits A et B pour donner

1 , Ø	registre	0	mémoire	dans registre
A , $\overline{A}$				
B , $\overline{B}$	registre	0	adresse	dans registre
$A \vee B$, $\overline{A} \wedge \overline{B}$				
$A \vee \overline{B}$, $\overline{A} \wedge B$	registre	0	mémoire	dans mémoire
$\overline{A} \vee \overline{B}$, $A \wedge B$				
$\overline{A} \vee B$, $A \wedge \overline{B}$	registre	0	mémoire	dans mémoire et registre
$A \oplus B$, $\overline{A \oplus B}$				

- tests logiques sur le contenu d'un registre avec comme masque

la quantité adressée	sans changer	sans sauter
la quantité adressée et ses moitiés échangées	en mettant à Ø	en sautant toujours
	en mettant à 1	en sautant si les bits masqués sont tous Ø
l'adresse appliquée à gauche	en complémentant	en sautant si les bits masqués ne sont pas tous Ø
l'adresse appliquée à droite	les bits du registre ainsi masqués	l'instruction suivante

- décalages et rotations algébriques ou logiques dans un registre ou dans un attelage registre,, registre + 1 d'un nombre de bits Ø (à gauche) ou Ø (à droite) égal à la quantité immédiate que constitue l'adresse (modulo 256).

Les opérations de groupage et dégroupage de caractères peuvent être faites plus simplement avec les instructions optionnelles de manipulation de caractères par pointeur.

1) - opérations d'entrées-sorties

- manipulation des registres d'état de l'équipement adressé

. écriture dans les registres de l'équipement adressé, cette instruction initialise certains registres, attribue les numéros des lignes d'interruptions à utiliser et sert à commander l'équipement (registre de fonction)
quantité immédiate = adresse sur la VOES

. lecture de l'état des registres de l'équipement
VOES position de mémoire adressée

. test et saut si les bits d'états masqués par la quantité immédiate sont tous nuls,
ne sont pas tous nuls.

- manipulations des données

Le principe est le même que pour manipuler les états, pour certains équipements faisant les échanges de données par un canal à accès direct, les instructions suivantes servent également aux états.

+ transfert VOES mémoire adressée
+ transfert mémoire adressée VOES
+ transfert sur pointeur incrémenté VOES mémoire
et test d'épuisement mémoire VOES

Les deux instructions de transfert sur pointeur incrémenté sont exécutées différemment selon qu'elles sont effectuées en mode normal ou comme première instruction d'une interruption prioritaire (instruction placée à l'adresse implicite d'entrée pour ce niveau de priorité). Dans le dernier cas si le compte de mots du pointeur

(situé en mémoire à l'adresse donnée par l'instruction) après incrémentation des deux moitiés, et transfert n'est pas nul, l'interruption cesse immédiatement et automatiquement, s'il est nul, l'instruction suivante est exécutée prioritairement, cette dernière doit sauver l'état arithmétique et le compteur programme de l'unité centrale pour permettre de poursuivre le travail interrompu plus tard.

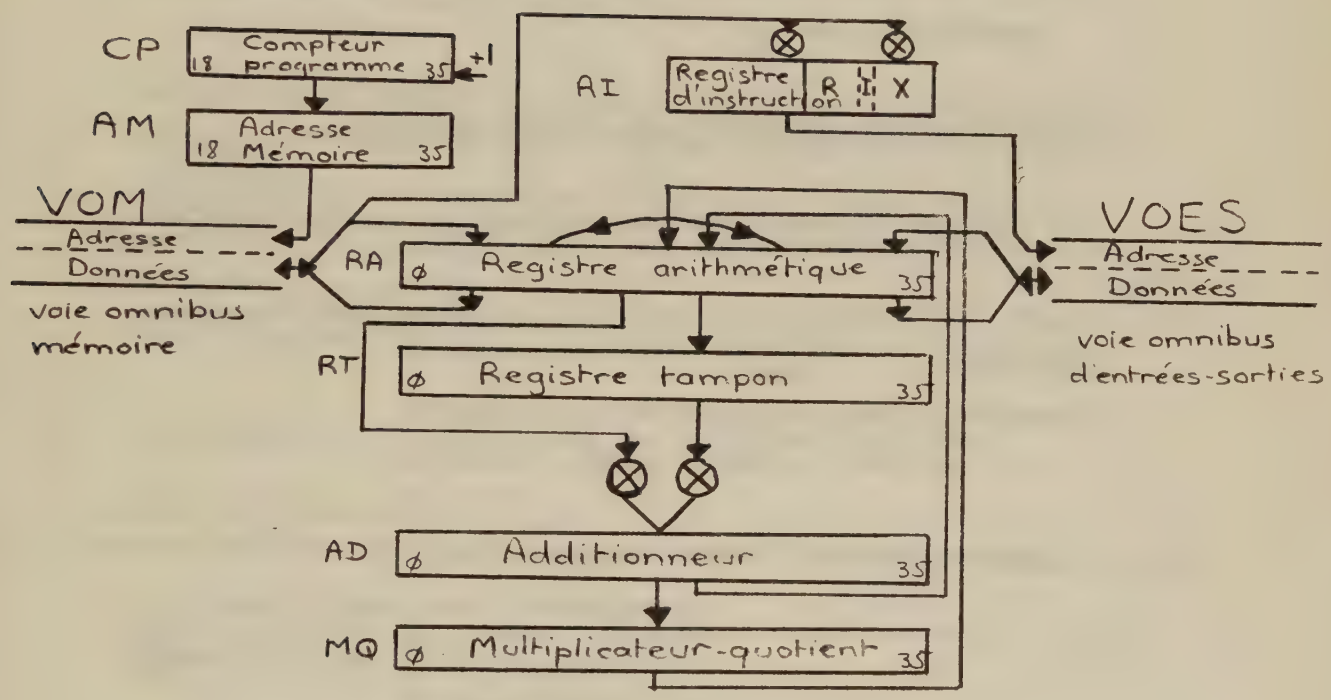
Une instruction exécutée prioritairement est exécutée en mode normal, l'ancien mode est mémorisé le temps de cette instruction pour être sauvé avec le compteur programme.

La simplicité de chaque instruction, leur regroupement possible par modules, les modes communs de rangement, de décision, entre familles différentes font qu'avec peu de circuits une unité centrale d'usage général est réalisable tout en satisfaisant la condition essentielle de disponibilité pour les échanges entre des équipements périphériques rapides et la mémoire centrale sans canal autonome. La réservation, le temps d'un transfert d'enregistrement d'un niveau d'interruption exclusivement réservé à un équipement permet sans effet sensible sur le programme interrompu de faire faire par l'unité centrale tout le travail d'un canal sélectif. (Le temps d'instruction est seul demandé, il n'y a pas de sauvegarde à faire, le mécanisme ne nécessite que 4 accès mémoire, là où un canal en nécessiterait un seul).

II - PRINCIPES DE FONCTIONNEMENT DE L'UNITE CENTRALE

1 - Les registres principaux et leurs connexions

Le contenu des registres est constitué d'états durables.



Aucun registre n'est adressable, seul le compteur programme et les drapeaux d'état arithmétique sont sous le contrôle d'instructions. Les registres de travail sont remis à zéro au début du décodage de chaque instruction. Les registres de contrôle comme l'adresse

mémoire peuvent conserver une information non vitale entre 2 instructions consécutives ; en cas de demande d'interruption, la première instruction pourra être reprise le compteur programme n'ayant pas été incrémenté, cela suppose que tous les résultats utiles ont été rangés en mémoire ou dans les registres adressables, cela arrive

- pour l'adressage indirect (à tous les niveaux),
- pour l'instruction XCT (à tous les niveaux),
- pour le transfert multiple,
- pour les codes non disponibles ou d'appel à adresse fixe translatable ou non.

2 - Fonctionnement asynchrone de l'unité centrale

On distingue 5 temps dans le déroulement d'une instruction.

1°/ - l'accès à l'instruction au moyen de AM chargé par CP sauf cas particuliers.
La mémoire délivre la quantité 1 mot dans RA
et pour la partie gauche dans RI

2°/ - le calcul d'adresse RA passe dans RT
le registre d'index est lu dans RA
l'addition est faite par AD replaçant le résultat dans RA

En cas d'indirection, RA passe dans AM et une nouvelle lecture a lieu dans RA et dans RI 13-17 (préservant le code instruction et le numéro de registre),
le calcul d'adresse recommence.

3°/ - l'accès aux opérandes nécessaires, à ce stade, il y a selon le code de l'instruction, éventuellement :

. lecture de la position adressée par RA passé dans AM
le résultat allant dans RA puis RT

. lecture du registre général dans RA

. lecture du registre suivant (modulo 16) (attelages) dans RA
après passage de RA dans MQ au travers de AD
après quoi MQ remplace RA qui remplace MQ au traver de AD
(échange pour que le premier registre retrouve
sa place dans RA)

. lecture de la position de mémoire pointée par la partie droite du registre dans RA ou gauche (après échange des 2 parties de RA). RA passe pour adressage dans AM et est conservé dans MQ au travers de AD après lecture de la quantité pointée dans RA, RA et MQ sont échangés via AD.
Dans la plupart des cas, le compteur programme est incrémenté.

4°/ - l'exécution de l'instruction, les opérandes étant dans

RA, RT et MQ
registre mémoire second registre ou quantité pointée par ler

AD et les portes de communication entre registres sont positionnées en fonction du code opération.

Selon le type d'instructions, l'exécution utilise plus ou moins d'impulsions donc, de temps, certaines font appel à des routines câblées dont l'exécution dépend des opérandes eux-mêmes (décalages, multiplications, divisions).
En cas de rupture de séquence, le compteur programme est lu ou/ et modifié selon les cas.

5°/ - le rangement des résultats

à ce stade, il y a selon le code d'instruction, éventuellement et dans l'ordre :

. écriture de RA dans le registre général indiqué dans RI

. écriture de RA ou bien de RT passé dans RA à l'adresse où fut faite éventuellement la dernière lecture, adresse conservée dans AM

. écriture de MQ passé dans RA dans le registre général suivant le registre (modulo 16.) indiqué dans RI

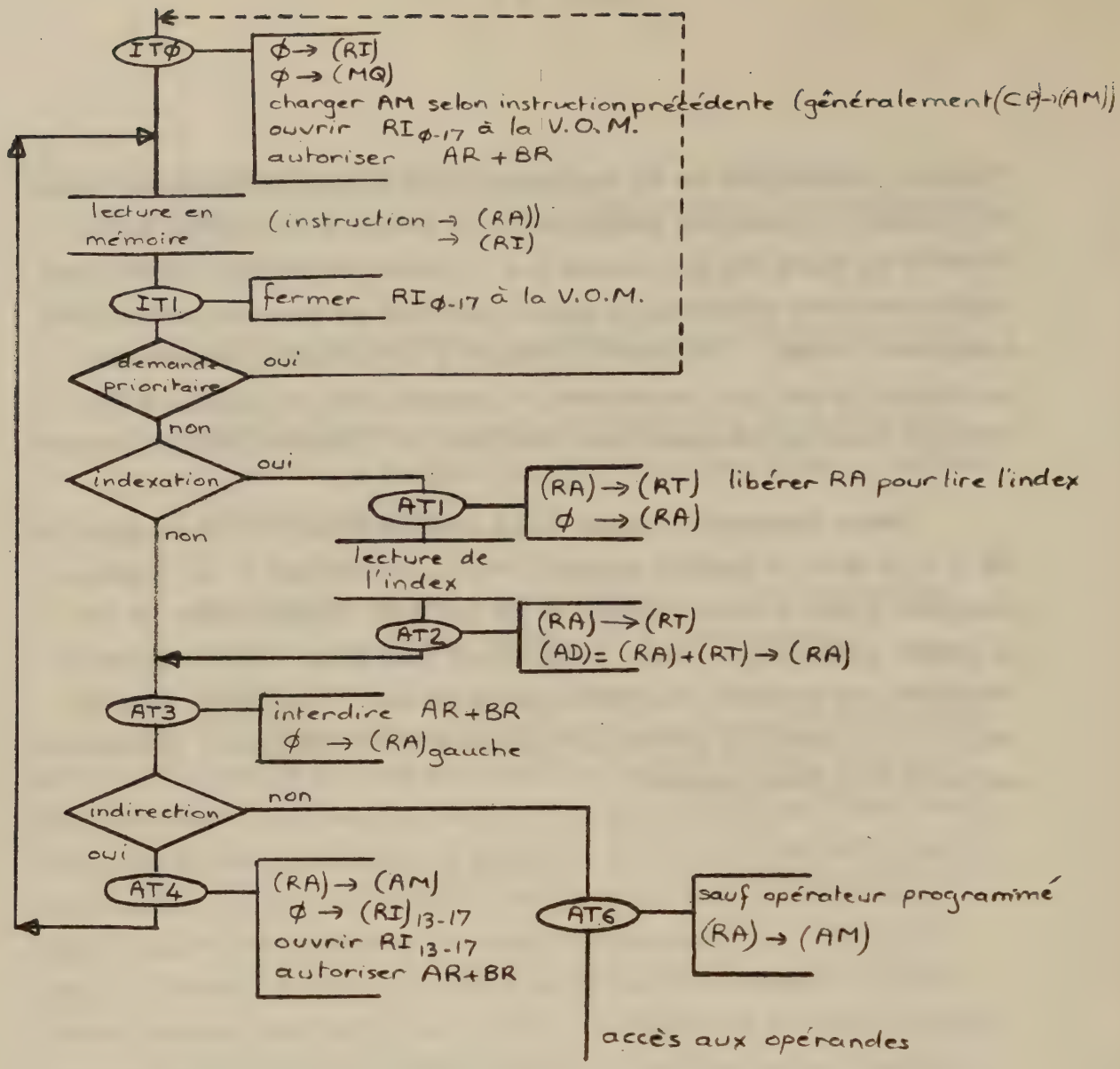
Si, à la fin de cette instruction, le registre arrêt-marche de l'unité centrale n'a pas été placé en position arrêt, soit par une instruction, (en mode normal), de saut inconditionnel avec arrêt nécessitant pour poursuivre la manoeuvre de la clé CONTINUER, soit par manoeuvre de la clé ARRET (ou de la clé REMISE A ZERO) soit momentanément pour exécuter une fonction commandée par clé de la console (EXAMEN, DEPOT, EXECUTION) entre deux instructions de programme au même titre qu'elles, alors l'enchaînement se fait automatiquement, en utilisant le contenu de CP (ou AM) pour aller chercher l'instruction suivante.

3 - Le mécanisme d'interruption de programme

Il est commandable par des instructions d'entrées-sorties à un équipement d'adresse particulière. Il est ainsi possible de préciser si le mécanisme est en action ou pas, si chaque priorité est possible ou non, et de placer par programme, une priorité en attente. Les demandes d'interruption sont acceptées et mises en attente à chaque accès mémoire (au moins un par instruction) sauf quand on fait une interruption prioritaire (qui dure 1 ou parfois 2 instructions avant sa

cessation automatique ou sa continuation en régime prioritaire, lequel dure jusque la cessation programmée). La priorité en attente devient priorité en cours dès que toutes les priorités en attente ou en cours supérieures sont terminées et que la priorité en cours de même niveau a également cessé. A ce moment (pendant l'interruption prioritaire) la priorité n'est plus en attente et aucune autre ne viendra à être acceptée avant le rangement des résultats de l'instruction prioritaire.

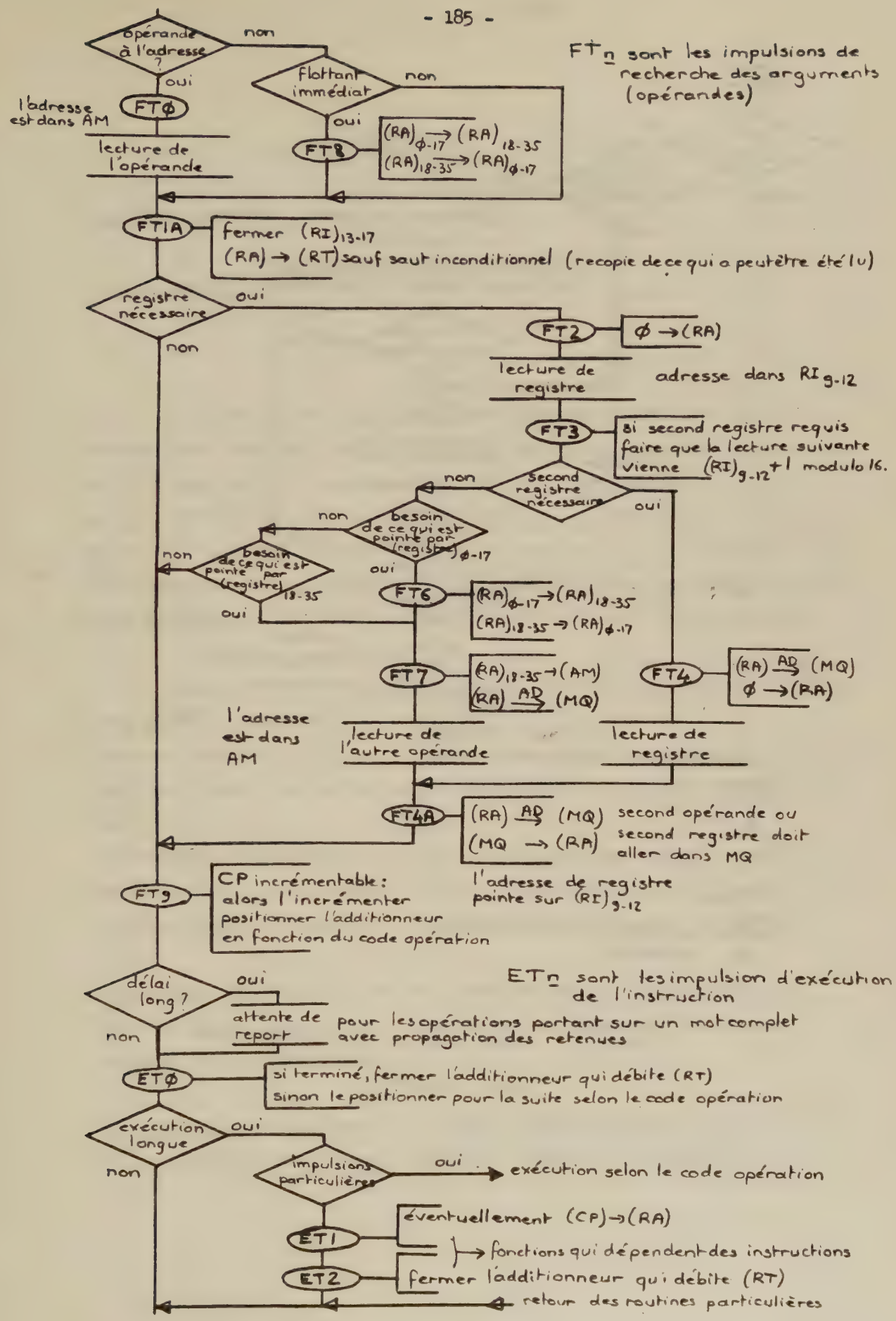
Cette instruction prise à une adresse dépendant de la priorité (4Ø + 2 n) doit en général sauver l'état arithmétique et le compteur programme à une adresse assurée de ne pas être écrasée avant la fin du régime prioritaire. Cette instruction peut aussi être un ordre de transfert sur pointeur incrémenté comme un canal simplifié, dans ce cas, il y a cessation automatique de la priorité sauf si l'instruction suivante doit être exécutée.

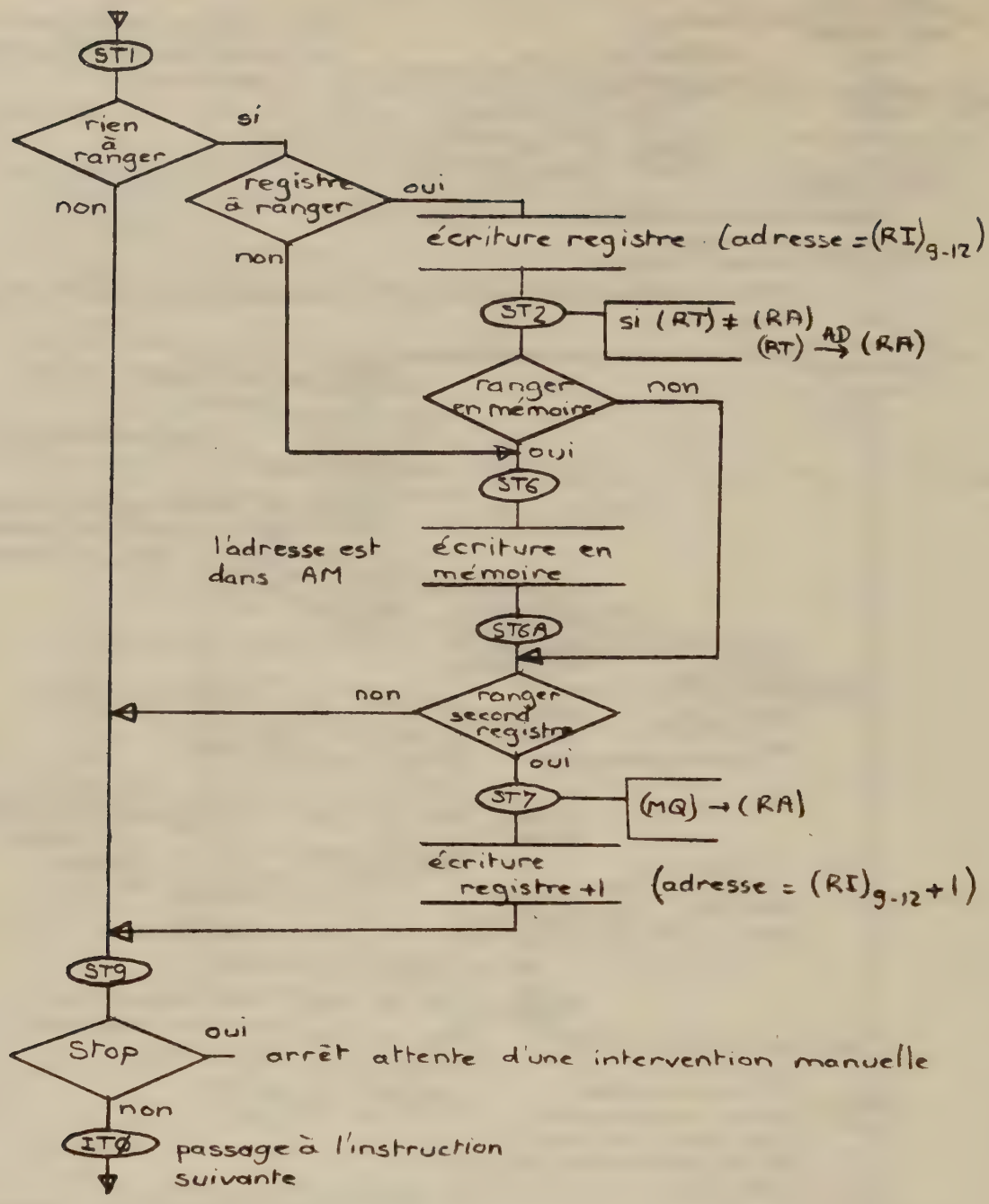


Séquences d'exécution d'une instruction

IT_n impulsions pour la recherche de l'instruction

AT_n impulsions pour le calcul de l'adresse effective (18 bits = indexation et indirection sans limite)


FTn sont les impulsions de recherche des arguments (opérandes)
opérande à l'adresse ?
non
oui
l'adresse est dans AM
FTØ
lecture de l'opérande
flottant immédiat
non
oui
FT8
(RA)Ø-17 → (RA)18-35
(RA)18-35 → (RA)Ø-17
FT1A
fermer (RI)13-17
(RA) → (RT) sauf saut inconditionnel (recopie de ce qui a peut être été lu)
registre nécessaire
oui
non
FT2
Ø → (RA)
lecture de registre
adresse dans RIg-12
FT3
si second registre requis faire que la lecture suivante vienne (RI)g-12 + 1 modulo 16.
second registre nécessaire
non
oui
besoin de ce qui est pointé par (registre)Ø-17
non
oui
besoin de ce qui est pointé par (registre)18-35
non
oui
FT6
(RA)Ø-17 → (RA)18-35
(RA)18-35 → (RA)Ø-17
FT7
(RA)18-35 → (AM)
(RA) AD (MQ)
FT4
(RA) AD (MQ)
Ø → (RA)
l'adresse est dans AM
lecture de l'autre opérande
lecture de registre
FT4A
(RA) AD (MQ)
(MQ → (RA)
second opérande ou second registre doit aller dans MQ
l'adresse de registre pointe sur (RI)g-12
FT9
CP incrémentable : alors l'incrémenter positionner l'additionneur en fonction du code opération
ETn sont les impulsion d'exécution de l'instruction
délai long ?
oui
non
attente de report
pour les opérations portant sur un mot complet avec propagation des retenues
ETØ
si terminé, fermer l'additionneur qui débite (RT)
sinon le positionner pour la suite selon le code opération
exécution longue
oui
non
impulsions particulières
oui
exécution selon le code opération
ET1
éventuellement (CP) → (RA)
fonctions qui dépendent des instructions
ET2
fermer l'additionneur qui débite (RT)
retour des routines particulières



ST_n impulsions de rangement des résultats

ANNEXE B

CONSTRUCTION D'UN SUPERVISEUR

Le mode de construction adopté doit présenter suffisamment de souplesse pour que chaque installation dispose vraiment du système jugé indispensable et conforme aux équipements dont elle dispose. La construction du superviseur doit être une tâche facile utilisant les outils normaux du système, ce qui suppose que l'on dispose déjà d'un superviseur en fonctionnement lequel peut être au début rudimentaire et inadapté à d'autres tâches particulières à l'installation.

Pour parvenir à ces fins, divers procédés ont été rassemblés :

- des modules bien définis et compatibles, avec pour ceux de gestion de périphériques, une présentation normalisée les rendant indépendants entre eux,

- un assemblage conditionnel du code de chaque module au moyen d'une liste commune de paramètres facilement modifiables,

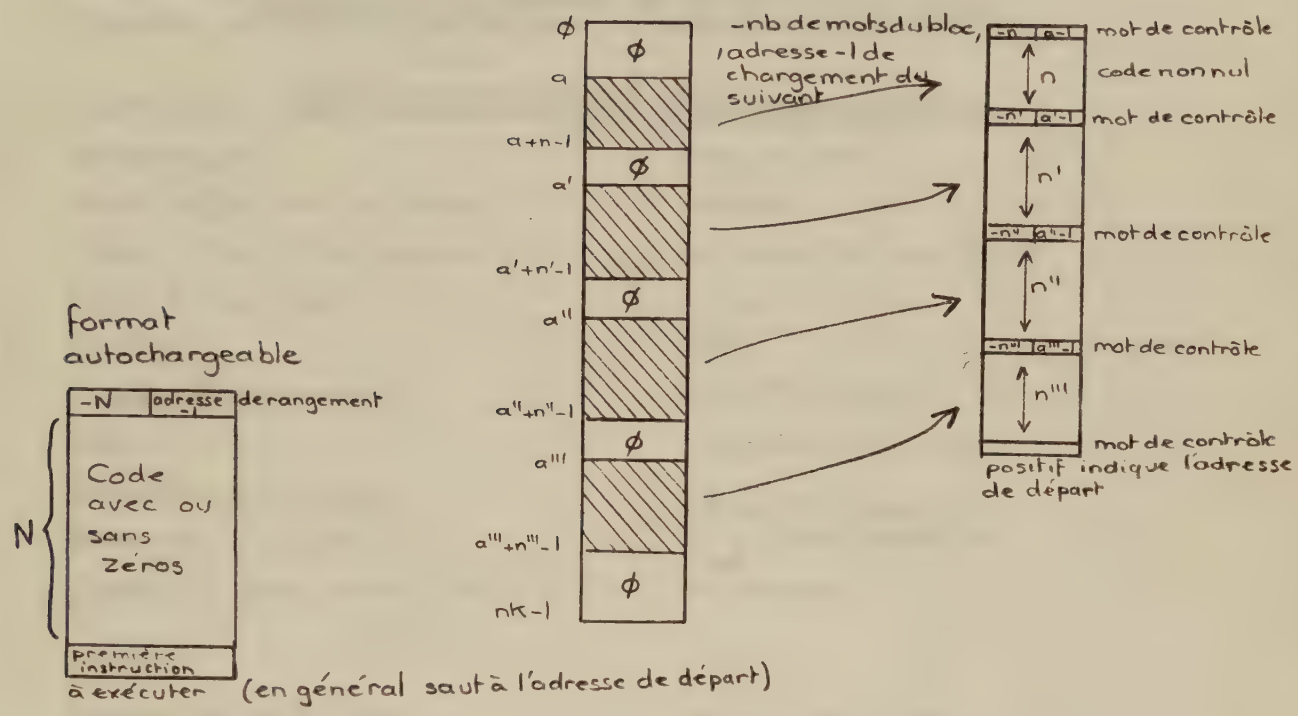
- une fusion ordonnée en une bibliothèque des différents modules dont on pense avoir besoin (fusion des résultats d'assemblages séparés de chaque module),

- une réunion et une liaison des seuls modules retenus compte tenu de la configuration, en un programme unique : le superviseur.

Certaines tables, certaines zones de rangement du superviseur sont de longueur variable, selon le nombre maximum d'utilisateurs ou le nombre de périphériques de chaque type. Ces paramètres n'influent pas sur le contenu (code) de chaque module mais servent à réserver des zones de rangement et à définir des constantes globales destinées à satisfaire des valeurs, non définies dans les autres modules, de symboles externes. Un petit programme questionneur peut générer une série de définitions de symboles en langage d'assemblage (fichier CONFIGURATION). Ce fichier assemblé avec un autre, généralement écrit sous forme conditionnelle en fonction du premier, permet de générer un module contenant toutes les zones variables, toutes valeurs non définissables dans les modules d'instructions et une série de requêtes permettant de choisir dans la bibliothèque des modules, ceux nécessaires au superviseur désiré. C'est le travail du programme "chargeur" de charger le premier module de tables et définitions globales, puis le reste du superviseur par choix dans la bibliothèque des modules, il vérifie que les symboles globaux ont partout la même valeur (compatibilité des modules assemblés séparément, entre eux) et qu'il n'y a pas de valeurs non définies (insuffisance de la bibliothèque ou insuffisance de déclarations globales dans certains modules). Le chargeur est capable d'éditer une carte de chargement donnant l'adresse des modules et de leurs étiquettes. Le chargeur, programme comme un autre, se détruit lui-même pour donner la place au programme qu'il a construit. Le superviseur ainsi bâti, comme tout programme normal, peut être recopié par un ordre au superviseur sur un support extérieur : bande magnétique.

Toutes les sorties de programme de la mémoire vers un support extérieur se font par le superviseur selon un format unique qui permet de façon générale d'économiser de la place en ne recopiant pas les mots nuls.

Ce format est dérivé du format autochargeable par action sur la touche d'autochargement de la console de la machine.



ANNEXE C

CORRESPONDANCE ENTRE LES EXPRESSIONS FRANÇAISES

ET LES EXPRESSIONS AMERICAINES ORIGINALES

Ce que j'entends par	Digital Equipment Co calls it :
VOM ou voie omnibus mémoire	Memory Bus
VOES ou voie omnibus d'entrée-sortie	I/O Bus
Canal à accès direct	Data Channel
Translation	Relocation
Violation de mémoire (de protection)	Illegal Memory Reference
Superviseur	Time Sharing Monitor
Autochargement	Read In Mode
Réinitialisation de disque	Refresh
Table d'Etat d'Equipement	DDB = device data block
Table d'Aiguillage	Dispatch table
Horloge	Clock
Console d'utilisateur	TTY = "Teletype"
Ordre	Command (Monitor mode)
Opération non câblée	UUO = Unimplemented User Operation
Opérateur programmé	
Numéro de tâche, numéro de travail	job number
Travail	job
Abandonner le système	to kill one's job
Désaffectation de console = abandon de travail	Detach (TTY)
Réaffectation de console	Attach (TTY)
Attente d'exécution d'ordre	Command Wait
Compteur général d'ordres en attente	COMCNT (command counter)
Caractère terminateur de ligne	Break character
Compte de lignes	TISYNC
Attente programmée d'entrée console	TIOW (TTY I/O Wait)
Utilisateur nul, travail d'attente	Nul Job
Sortie console en cours	Type Out In Progress (TOIP)
Démission d'interruption prioritaire	Dismissal of an Interrupt

Ce que j'entends par	Digital Equipment Co calls it :
Tampon	Buffer
Enchaînement des travaux	Scheduling
Décodeur d'opérateur programmé	UUOCON (UUO Control)
Utilisateur courant	Current job (his job number is in location JOB)
Etat du système (ordre)	Status (COMCON)
Attribuer (ordre)	ASSIGN (COMCON)
Mémoire (ordre)	CORE (COMCON)
Recherche (ordre)	GET (COMCON)
Lancement (ordre)	START (COMCON)
Arrêt (ordre)	HALT or ↑C (COMCON)
Sauvetage (ordre)	SAVE (COMCON)
Arrivée (ordre)	LOGIN (COMCON)
Programme créateur de fichier	TECO (or EDITOR)
Décodage et exécution des ordres	COMCON (Command Control)
Recherche et lancement (ordre)	RUN (COMCON)
Transfert en cours	no DDB with IOACT in DEVIOS for all channel inited by this job
Zone mémoire immobile	SHF = Ø in JOB Status (JBTSTS) et SWP = Ø
Equipement	Device
Fichier	File
Nom de fichier (1 mot et demi)	Name + Extension (6 + 3 Sixbit characters)
GESTION DE LA Mémoire	COREI (memory allocation)
Mémoire réelle	CORE
Mémoire virtuelle	Virtual core, virtual memory Swapping space
Exécution suspendue (ordres)	delayed command (DLYCOM)
Rentrée en mémoire réelle	Swap-in
Echange mémoire réelle - mémoire virtuelle	Swapping
Gestion des programmes réentrants	SEGCON (Segment control)
Gestion des programmes partagés	
Départ (ordre)	KJOB (COMCON)

Ce que j'entends par	Digital Equipment Co calls it :
Gestion des tours de passage	SCHED (scheduling)
Disque rapide à tête fixe	Fixed Head Disk (Burroughs disk)
Fragments en mémoire virtuelle	Fragmentation in swapping space
Zone banalisée du superviseur	Free Core (DDBTAB)
Compteur de K disponibles (ou reste)	Tally
pour K réels	CORTAL (Core-tally)
pour K virtuels	VIRTAL (Virtual-space-tally)
Queue d'attente spéciale pour exécution différée d'un ordre	Command Wait Queue (CMWQ)
Expansion en retour de mémoire virtuelle	EXPAND (JXPN in JBTSTS)
Dégager (engager) les K occupés de la table d'occupation	CORSTG routine, updating CORTAB (Core allocation table)
Recherche de trou dans la table d'occupation	HOLSRC (routine within COREI)
Table d'adresse du segment	JBTADR table
Taille du plus gros trou	BIGHOL (location)
Adresse du trou (table d'occupation)	HOLEF (hole-flag)
Tassement (en mémoire réelle) = déplacement des trous vers le haut	Schuffling
Allocation de mémoire par opérateur programmé	CORE UUO (CALLIAC, II)
Attente = réarrangement des queues	WSCHED called by a PUSHJ to force a rescheduling
Ajuster les registres de base (routine)	SETREL
Travail arrêté	Job in STOP QUEUE (STOPQ) RUN bit is Ø in Status (JBTSTS)
Bloc (mémoire virtuelle, fichiers)	Logical block (128. words) on disk-like devices
Taille maximum autorisée pour un travail	CORMAX (location)
Segment sélectionné pour tassement	SHFWAT (shuffle waiting job)
Echange entre mémoires en cours	SQREQ (swapping queue request) (flag)

Ce que j'entends par	Digital Equipment Co calls its :
Transfert à acquitter	FINISH (location)(SWAP)
Segment sélectionné pour rentrer	FIT (location)(SWAP)
Expansions (compteur)	XJOB (expanding job counter)
Routine pour essayer de bouger un segment	CHKSHF (check-shuffling)
Segment sélectionné pour sortir	FORCE (location)(SWAP)
Calcul de parité global (avant sortie en mémoire virtuelle)	Folded checksum (in JBTCHK table)
Répertoire	Directory - User file-directory on disk (UFD)
Temps de protection en mémoire réelle	In-core-protect-time
Avoir de temps	Quantum time (in JBTSTS)
Changements de queue différés	QJOB counter
Table des équipements partageables devenus disponibles	AVALTB table for sharable devices
Drapeau de disponibilité	location within AVAL TB
QP1, QP2, QP3, queues de passage	PQ1, PQ2, PQ3 Processor-Queues
Initialisation d'équipement	INIT (UUO) ou OPEN
Initialisation d'un fichier :	
ouverture en lecture ouverture en écriture	LOOKUP (UUO) ENTER (UUO)
Etat du transfert	FILE STATUS
Fermeture de fichier	CLOSE (UUO)
Modification de référence de fichier	RENAME (UUO)
Abandon d'équipement	RELEAS (UUO)
Ordre de réservation et d'appellation symbolique	ASSIGN (COMMAND)
Canal programmé	Channel
Lecture directe	Dump input
Lecture avec tampons	Buffered input
Demande de construction de tampons	INBUF OUTBUF (UUO)
Première adresse libre dans la zone de l'utilisateur	JOBFF (JOBDAT) first-free location

Ce que j'entends par	Digital Equipment Co calls it :
Opérateur de remise à zéro logique de toutes les entrées-sorties	CALLI Ø (UUO) ou RESET (UUO)
Indicateur de remplissage	USE bit (buffer)
Indicateur de premier accès	RING USE BIT (buffer header)
Module de synchronisation	WSYNC (subroutine)
Module d'attente de fin d'activité	WAITI (subroutine)
Blocs d'identification	RIB (Retrieval Information Block)
Fichier répertoire général	MFD (Master File Directory)
Table d'occupation de disque	SAT (Storage Allocation Table)
Cellule d'accès	Acces Table
Code d'accès	Protection code (9 bits)
Implantation variable	Relocatability

ANNEXE D

LISTES DU MONITEUR

```
BASMON  PARTIE BASSE DU MONITEUR DE PAGINATION  09.08.78        MACRO.V43 10:49  2-SEP-78 PAGE 1
CB    .MAC      DECLARATIONS

                                 ;CETTE FILE A ETE ASSEMBLEE AVEC LA FILE DES PARAMETRES COMMUNS
                                 ;AUX SOUS-PROGRAMMES DE PAGINATION  CB.MAC

                                 XLIST
                                 LIST
                                 TITLE   BASMON  PARTIE BASSE DU MONITEUR DE PAGINATION  09.08.78
                                 SUBTTL  DECLARATIONS

                                 ,ASSEMBLAGE AVEC LA FILE DES PARAMETRES CB.MAC

                                 INTERN  ABSMAX  ,MEMAPR ,ACCUS  ,ALLCAN ,APRORP ,ATEMPR ,CPOPJ  ,PAGEPC
                                 INTERN  CPOPJ1  ,FORTY  ,MAXBLK ,MEM41  ,SEGADR ,SEGLOC ,MEMBLP ,MEMODT
                                 INTERN  MEMFIL  ,MEMREN ,MEMSA  ,MEMSYM ,SEGPAG ,MODUMP ,PADRDX ,PAGBLN
                                 INTERN  PAGADR  ,PAGBLK ,PAGBLP ,PAGWTR ,PAGLNG ,PILPTR ,PILSAV ,PINSAC
                                 INTERN  PINSCD  ,PINSDX ,PTACAC ,PURARG ,PURWWA ,PURWWI ,PURWWO ,PURWWZ
                                 INTERN  PUROPE  ,PURUUO ,PUUOAC ,PUUOPE ,SVTAC  ,SVTAC2 ,TACSAV ,TTYWWI
                                 INTERN  UWWTAB  ,US141  ,US1REN ,US1SA  ,US1DDT ,USRSAV ,UTILNG ,ZTEMPR
                                 INTERN  UUO0    ,MEMFF  ,USERGO ,US1APR ,CORIOW ,PILE   ,PAGZDR
                                 INTERN  SEGBLK  ,SEGZDR ,MEMBLK

                                 EXTERN  USR41   ,USRREL ,USRAPR ,USRREN ,USRSA  ,USRDDT ,USERGB ,TAPMES
                                 EXTERN  UUOINT  ,US1AP1 ,CHKFRG

                                 INTERN  JOBUUO  ,JOB41  ,JOBDDT ,JOBREN ,JOBSA  ,JOBFF  ,JOBSYM ,JOBAPR

                        000040   JOBUUO=40
                        000041   JOB41=41
                        000074   JOBDDT=74
                        000124   JOBREN=124
                        000120   JOBSA=120
                        000121   JOBFF=121
                        000116   JOBSYM=116
                        000125   JOBAPR=125

                                 HLON


                                 SUBTTL  COMMUTATION DES MODES USAGER EN USAGER DE PAGE

                                 ,ROUTINE DE PASSAGE DU MODE UTILISATEUR AU MODE INTERMEDIAIRE:SAUVETAGE
                                 ,DES ACCUMULATEURS ET ETABLISSEMENT DU POINTEUR DE PILE

000000' 000000  000000  USRSAV: Z                       ;ICI PAR JSR
000001' 202740  000120'         MOVEM   17,ACCUS+17     ;D'ABORD LE PLUS ELEVE
000002' 201740  000101'         MOVEI   17,ACCUS        ;POINTEUR DE TRANSFERT
000003' 251740  000117'         BLT     17,ACCUS+16     ;TRANSFERT DES 15(10) AUTRES AC'S
000004' 200740  000047'         MOVE    PIL,PILPTR      ;PONTEUR DE PILE
000005' 047000  777775          CALLI   UUOON,UUOSW     ;RETABLIR LES UUOS
000006' 047140  777775          CALLI   HGHON,HIGHSW    ;RETABLIR LE SEGMENT HAUT
000007' 254120  000000'         JRSTF   @USRSAV         ;RETOUR D'UN JSR

000010' 260740  000000* USERGO: PUSHJ   PIL,CHKFRG      ;VERIFIER QUE LE SEGMENT LOCAL A ETE MIS DANS LA PAGE
000011' 200040  000123'         MOVE    TAC,APRORP      ;TRAPPES AUTORISEES PAR L'UTILISATEUR
000012' 435040  020000          IORI    TAC,ILM         ;ASSURER AU MOINS L'ILM
000013' 047040  000016          CALLI   TAC,ILMENB      ;AUPRES DU MONITEUR
000014' 047100  777775          CALLI   HGHOFF,HIGHSW   ;SUPPRIMER LE SEGMENT HAUT
000015' 047040  777775          CALLI   UUOOFF,UUOSW    ;SUPPRIMER LES UUOS
                                DEFINE  POINTE(A) ,<IRP A
                                MOVEI   TAC,US1'A
                                HRRM    TAC,JOB'A>>
                                POINTE  <41,REN,SA,APR>*IRP
000016' 201040  000031'         MOVEI   TAC,US1'41
000017' 542040  000041          HRRM    TAC,JOB'41
000020' 201040  000050'         MOVEI   TAC,US1'REN
000021' 542040  000124          HRRM    TAC,JOB'REN
000022' 201040  000055'         MOVEI   TAC,US1'SA
000023' 542040  000120          HRRM    TAC,JOB'SA
000024' 201040  000034'         MOVEI   TAC,US1'APR
000025' 542040  000125          HRRM    TAC,JOB'APR
000026' 205740  000101'         MOVSI   17,ACCUS        ;RESTAURER LES AC'S DE L'UTILISATEUR
000027' 251740  000017          BLT     17,17           ;...
000030' 254120  000100'         JRSTF   @PAGEPC         ;ET RETOUR A L'UTILISATEUR
                                ,ICI PAR JSR SIMULE OU NON DE JOB41 SUR UUO

000031'                         US141:
000031' 000000  000000  UUO0:   Z                       ;LE PC AU MOMENT DE L'UUO EST RANGE ICI
000032' 264000  000000'         JSR     USRSAV          ;SAUVER ET RETABLIR LA LIAISON AVEC LE HAUT
000033' 254000  000000*         JRST    UUOINT          ;VOIR LA ROUTINE GENERALE


                                ,ICI PAR SAUT DE JOBAPR SUR VIOLATION DE MEMOIRE

000034' 264000  000000' US1APR: JSR     USRSAV          ;SAUVER LES AC'S ET ACCEDER AU SEGMENT HAUT
000035' 254000  000000*         JRST    US1AP1          ;Y SAUTER MAINTENANT QUE C'EST POSSIBLE

                                ,POINTEURS DIVERS

                        000000  ZZ==INS=UUO
                                IFE     ZZ,<
000036'                         PINSAC:>
000036' 270400  000016  PUUOAC: POINT   4,UUO,12        ;LA ZONE AC DE CE QUI EST DANS L'AC UUO
                                IFN     ZZ,<
                                PINSAC: POINT   4,INS,12        ;IDEM POUR L'AC INS (INSTRUCTION)
                                >
                                IFE     ZZ,<
000037'                         PINSCD:>
000037' 331100  000016  PUUOPE: POINT   9,UUO,8         ;POINTEUR SUR LE CODE OPERATION (UUO)
                                IFN     ZZ,<
                                PINSCD: POINT   9,INS,8         ;POINTEUR SUR LE CODE OPERATION
                                >
000040' 220400  000016  PINSDX: POINT   4,INS,17        ;POINTEUR SUR LA ZONE INDEX
000041' 220400  000004  PADRDX: POINT   4,ADR,17        ;POINTEUR SUR LA ZONE INDEX
000042' 270400  000001  PTACAC: POINT   4,TAC,12        ;ZONE AC DE CE QUI EST DANS TAC

000043' 350017  000000  CPOPJ1: AOS     (PIL)           ;RETOUR A L'INSTRUCTION D'APPEL+2
000044' 263740  000000  CPOPJ:  POPJ    PIL,0           ;RETOUR A L'INSTRUCTION D'APPEL+1
000045' 777777  000045' CORIOW: IOWD    1,.+1
000046' 000000  000000          Z
000047' 777750  000207' PILPTR: IOWD    ASSEZ,PILE      ;POINTEUR INITIAL
                        000040  FORTY=JOBUUO
```

```
BASMON  PARTIE BASSE DU MONITEUR DE PAGINATION  09.08.78        MACRO.V43 10:49  2-SEP-78 PAGE 2
BASMON.MAC      COMMUTATION DES MODES USAGER EN USAGER DE PAGE

                                ;CORRESPONDANCES ENTRE LES ENTREES MONITEUR ET LES ENTREES UTILISATEUR

 000050' 264000  000000' US1REN: JSR     USRSAV          ;ETABLIR LA LIAISON AVEC LE HAUT
 000051' 550200  000000*         HRRZ    ADR,USRREN      ;ADRESSE DE RE-ENTREE
 000052' 326200  000000*         JUMPN   ADR,USERG0      ;SI SPECIFIEE Y ALLER APRES QQ PRECAUTIONS
 000053' 201000  000067'         MOVEI   [ASCIZ  ,?PAS DE REE.]
 000054' 254000  000000*         JRST    TAPMES          ;EXIT AVEC CE MESSAGE

 000055' 264000  000000' US1SA:  JSR     USRSAV          ;REATABLIR LA LIAISON AVEC LE HAUT
 000056' 550200  000000*         HRRZ    ADR,USRSA       ;ADRESSE DE DEPART
 000057' 326200  000052*         JUMPN   ADR,USERG0      ;SI SPECIFIEE
 000060' 201000  000072'         MOVEI   [ASCIZ  ,?PAS DE DEPART.]
 000061' 254000  000054*         JRST    TAPMES          ;EXIT AVEC CE MESSAGE

 000062' 264000  000000' US1DDT: JSR     USRSAV          ;RETABLIR LA LIAISON AVEC LE HAUT
 000063' 550200  000000*         HRRZ    ADR,USRDDT      ;ADRESSE DE DEPART
 000064' 326200  000057*         JUMPN   ADR,USERG0      ;SI SPECIFIEE ...
 000065' 201000  000075'         MOVEI   [ASCIZ  ,?PAS DE DDT.]
 000066' 254000  000061*         JRST    TAPMES          ;EXIT AVEC CE MESSAGE
                                 XLIST
                                 LIST

                                 SUBTTL  ZONES DE RANGEMENTS D'ECRITURE ET DE MEMOIRES...

 000100'                         ATEMPR:                 ;DEBUT DE LA ZONE TEMPORAIRE ...(ZTEMPR)
 000100'                         PAGEPC: BLOCK   1       ;COMPTEUR PROGRAMME DE REPRISE
 000101'                         ACCUS:  BLOCK   20      ;POUR LES AC'S UTILISATEUR
 000121'                         ALLCAN: BLOCK   2       ;TABLE DE CORRESPONDANCES DES CANAUX
 000123'                         APRORP: BLOCK   1       ;DRAPEAUX QUE L'UTILISATEUR ENTEND
                                                         ;TRAITER LUI MEME
 000124'                         SVTAC:  BLOCK   TAC2-TAC+1  ;POUR RANGER TAC...TAC2
                        000126'  SVTAC2=.-1
 000127'                         TACSAV: BLOCK   1       ;POUR SAUVER TAC
 000130'                         PILSAV: BLOCK   1       ;POUR SAUVER PIL
 000131'                         PURHHA:                 ;DEBUT DE LA ZONE DES TETES DE TAMPONS
 000131'                         PURHHI: BLOCK   3
 000134'                         PURHHO: BLOCK   3
                        000136'  PURHHZ=.-1
 000137'                         PURUUO:                 ;POUR METTRE LES UUOS A EXECUTER
 000137'                         PUROPE: BLOCK   3       ;POUR METTRE LES OPERATIONS A EXECUTER
 000142'                         PURARG: BLOCK   4       ;POUR LES ARGUMENTS D'UUO...
 000146'                         UHHTAB: BLOCK   17      ;POUR LES ADRESSES ABSOLUES DES TETES DE TAMPONS
 000165'                         TTYHHI: BLOCK   3       ;POUR LA LECTURE SUR TTY A L'INITIALISATION
 000170'                         BASLNG: BLOCK   1       ;LONGUEUR DE LA BASE
 000171'                         PAG0LN: BLOCK   1       ;LONGUEUR DE LA PAGE 0
 000172'                         PAGHTR:                 ;HAUTEUR D'UNE PAGE
 000172'                         PAGLNG:                 ;LONGUEUR D'UNE PAGE
 000172'                         UTILNG: BLOCK   1       ;NMB DE MOTS UTILES PAR PAGE
 000173'                         MAXBLK: BLOCK   1       ;NMB DE BLOCS DE DSK UTILISES
 000174'                         ABSMAX: BLOCK   1       ;ADRESSE ABSOLUE MAXIMUM TOLERABLE
 000175'                         PAGADR: BLOCK   1       ;ADRESSE ABSOLUE DE TETE DE LA PAGE EN COURS
 000176'                         PAGZDR: BLOCK   1       ;ADRESSE ABSOLUE DE FIN DE PAGE EN COURS
 000177'                         PAGBLK: BLOCK   1       ;NO DE BLOC CORRESPONDANT A CETTE ADRESSE
 000200'                         PAGBLP: BLOCK   2       ;IOWD DE MANIPULATION DE PAGE
 000202'                         MEMBLP: BLOCK   2       ;IOWD DE MANIPULATION DE BLOC
 000204'                         SEGPAG: BLOCK   1       ;ADRESSE ABSOLUE DE TETE DE LA PAGE DE CE BLOC
 000205'                         SEGADR: BLOCK   1       ;PREMIERE ADRESSE ABSOLUE DE CE BLOC
 000206'                         SEGZDR: BLOCK   1       ;ADRESSE ABSOLUE DE FIN DU SEGMENT
 000207'                         SEGBLK: BLOCK   1       ;NO DE BLOC DU SEGMENT EN COURS
 000210'                         PILE:   BLOCK   <ASSEZ-*30>  ;PILE DE TRAVAIL
 000240                          MODUMP: BLOCK   3       ;POUR OUVRIR LA FILE D'EXTENSION MEMOIRE
 000243'                         MEMFIL: BLOCK   4       ;POUR OUVRIR LA FILE MEMOIRE
 000247'                         SEGLOC:
 000247'                         MEMBLK: BLOCK   BLKSIZ  ;BLOC PARTIEL DE MEMOIRE
                        000310'  MEM41=  MEMBLK+JOB41
                        000343'  MEMDDT= MEMBLK+JOBDDT
                        000373'  MEMREN= MEMBLK+JOBREN
                        000374'  MEMAPR= MEMBLK+JOBAPR
                        000367'  MEMSA=  MEMBLK+JOBSA
                        000370'  MEMFF=  MEMBLK+JOBFF
                        000365'  MEMSYM= MEMBLK+JOBSYM
                        000446'  ZTEMPR=.-1
                                 END

NO ERRORS DETECTED

PROGRAM BREAK IS 000447
```

```
BASMON  PARTIE BASSE DU MONITEUR DE PAGINATION  09.08.70        MACRO.V43 10:49  2-SEP-70 PAGE 3
BASMON.MAC      SYMBOL TABLE

ABSMAX          000174' INT     ACCUS           000101' INT     ACU             000010  SPD     ADR             000224  SPD
ALAPRB          000003  SPD     ALLCAN          000121' INT     APRDRP          000123' INT     ASCIIL          000201  SPD
ASSEZ           000030  SPD     ATEMPR          000100' INT     BASLNG          000170'         BKSHFT  777777  777771  SPD
BLKSIZ          000200  SPD     CHKFRG          000010' EXT     CHN             000006  SPD     CORE            000011  SPD
CORIOW          000045' INT     CPOPJ           000044' INT     CPOPJ1          000043' INT     DAT             000205  SPD
DDTOUT          000003  SPD     DEVCHR          000001  SPD     DRP             000000  SPD     DUMP            000217  SPD
EXIT            000012  SPD     FORTY           000040  INT     HGWOFF          000002  SPD     HGWON           000203  SPD
HIGHSW  777777  777775  SPD     HORPAG          000004  SPD     IDX             000011  SPD     ILM             000228  SPD
ILMENB          000016  SPD     IMMB            000001  SPD     INS             000016  SPD     JDABLK          000201  SPD
JOB41           000041  INT     JOBAPR          000125  INT     JOBDDT          000074  INT     JOBFF           000121  INT
JOBPAG          000140  SPD     JOBREN          000124  INT     JOBSA           000120  INT     JOBSYM          000116  INT
JOBUUO          000040  INT     LOALOC          400000  SPD     LOC             000007  SPD     MAXBLK          000173' INT
MEM41           000310' INT     MEMAPR          000374' INT     MEMBLK          000247' INT     MEMBLP          000202' INT
MEMDDT          000343' INT     MEMFF           000370' INT     MEMFIL          000243' INT     MEMORY          000200  SPD
MEMREN          000373' INT     MEMSA           000367' INT     MEMSYM          000365' INT     MODUMP          000240' INT
NADRB           000002  SPD     NORPAG          000010  SPD     OPTESS  777777  777777  SPD     PADROX          000341' INT
PAGØLN          000171' INT     PAGADR          000175' INT     PAGBLK          000177' INT     PAGBLP          000200' INT
PAGEPC          000100' INT     PAGHTR          000172' INT     PAGLNG          000172' INT     PAGLOC          000000  SPD
PAGZDR          000176' INT     PEEK            000033  SPD     PIL             000017  SPD     PILE            000210' INT
PILPTR          000047' INT     PILSAV          000130' INT     PINSAC          000036' INT     PINSCD          000037' INT
PINSDX          000040' INT     PTACAC          000042' INT     PURARG          000142' INT     PURHWA          000131' INT
PURHWI          000131' INT     PURHWO          000134' INT     PURHWZ          000136' INT     PUROPE          000137' INT
PURUUO          000137' INT     PUUOAC          000036' INT     PUUOPE          000037' INT     SD              000015  SPD
SEGADR          000205' INT     SEGBLK          000207' INT     SEGLOC          000247' INT     SEGPAG          000204' INT
SEGZDR          000206' INT     SETDDT          000002  SPD     SLEEP           000031  SPD     SVTAC           000124' INT
SVTAC2          000126' INT     TAC             000001  SPD     TAC1            000002  SPD     TAC2            000003  SPD
TACSAV          000127' INT     TAPMES          000066' EXT     TTY             000000  SPD     TTYHWI          000165' INT
UHHTAB          000146' INT     US141           000031' INT     US1AP1          000035' EXT     US1APR          000034' INT
US1DDT          000062' INT     US1REN          000050' INT     US1SA           000055' INT     USERGØ          000264' EXT
USERGO          000010' INT     USR41           000000  EXT     USRAPR          000000  EXT     USRDDT          000063' EXT
USRREL          000000  EXT     USRREN          000051' EXT     USRSA           000056' EXT     USRSAV          000228' INT
UTILNG          000172' INT     UUO             000016  SPD     UUOØ            000031' INT     UUOINT          000033' EXT
UUOOFF          000001  SPD     UUOON           000000  SPD     UUOSW   777777  777775  SPD     ZTEMPR          000446' INT
ZZ              000000  SPD

2K CORE USED


ABSMAX      11     160#
ACCUS       11      44      45      46      70     137#
ACU          5#
ADR          5#     102     114     115     120     121     126     127
ALAPRB       5#
ALLCAN      11     138#
APRDRP      11      53     139#
ASCIIL       5#
ASSEZ      109     170#
ATEMPR      11     135#
BASLNG     154#
BKSHFT       5#
BLKSIZ       5#     174
CHKFRG      22#      52
CHN          5#
CORE         5#
CORIOW      18     107#
CPOPJ       11     106#
CPOPJ1      12     105#
DAT          5#
DDTOUT       5#
DEVCHR       5#
DRP          5#
DUMP         5#
EXIT         5#
FORTY       12     110#
HGWOFF       5#      56
HGWON        5#      49
HIGHSW       5#      49      56
HORPAG       5#
IDX          5#
ILM          5#      54
ILMENB       5#      55
IMMB         5#       5
INS          5#      88     101
JDABLK       5#
JOB41       24      27#      63     175
JOBAPR      24      33#      69     178
JOBDDT      24      28#     176
JOBFF       24      31#     180
JOBPAG       5#
JOBREN      24      29#      65     177
JOBSA       24      30#      67     179
JOBSYM      24      32#     181
JOBUUO      24      26#     110
LOALOC       5#
LOC          5#
LSTCB        5
MAXBLK      12     159#
MEM41       12     175#
MEMAPR      11     178#
MEMBLK      19     174#     175     176     177     178     179     180     181
MEMBLP      12     165#
MEMDDT      12     176#
MEMFF       18     180#
MEMFIL      13     172#
MEMORY       5#
MEMREN      13     177#
MEMSA       13     179#
MEMSYM      13     181#
MODUMP      13     171#
NADRB        5#      5
NORPAG       5#
OPTESS       5
PADROX      13     102#
PAGØLN      13     155#
PAGADR      14     161#
PAGBLK      14     163#
PAGBLP      14     164#
PAGEPC      11      72     136#
PAGHTR      14     156#
PAGLNG      14     157#
PAGLOC       5#
```

PAGZDR	18	162#												
PEEK	5#													
PIL	5#	47	52	105	106									
PILE	18	109	170#											
PILPTR	14	47	109#											
PILSAV	14	144#												
PINSAC	14	90#												
PINSCD	15	96#												
PINSDX	15	101#												
PTACAC	15	103#												
PURARG	15	151#												
PURHHA	15	145#												
PURHHI	15	146#												
PURHHO	15	147#												
PURHHZ	15	148#												
PUROPE	16	150#												
PURUUO	16	149#												
PUUOAC	16	91#												
PUUOPE	16	97#												
SQ	5#													
SEGADR	12	167#												
SEGBLK	19	169#												
SEGLOC	12	173#												
SEGPAG	13	166#												
SEGZDR	19	168#												
SETDDT	5#													
SLEEP	5#													
SVTAC	16	141#												
SVTAC2	16	142#												
TAC	5#	5	53	54	55	62	63	64	65	66	67	68	69	103
	141													
TAC1	5#													
TAC2	5#	141												
TACSAV	16	143#												
TAPMES	21#	117	123	129										
TTY	5#													
TTYHWI	16	153#												
UHHTAB	17	152#												
US141	17	62	75#											
US1AP1	22#	84												
US1APR	18	68	83#											
US1DDT	17	125#												
US1REN	17	64	113#											
US1SA	17	66	119#											
USERG0	21#	115	121	127										
USERGO	18	52#												
USR41	21#													
USRAPR	21#													
USRDDT	21#	126												
USRREL	21#													
USRREN	21#	114												
USRSA	21#	120												
USRSAV	17	43#	50	77	83	113	119	125						
UTILNG	17	158#												
UUO	5#	5	88	91	97									
UUO0	18	76#												
UUOINT	22#	78												
UUOOFF	5#	57												
UUOON	5#	48												
UUOSW	5#	5	46	57										
ZTEMPR	17	182#												
ZZ	88#	89	92	95	98									

```
MONDAT  JOBDATA AREA EN PAGINATION      MACRO.V43 10:48  2-SEP-78 PAGE 1
CB    .MAC      REDEFINITION DE CERTAINES ADRESSES DE JOBDAT.REL

                                ;CETTE FILE A ETE ASSEMBLEE AVEC LA FILE DES PARAMETRES COMMUNS
                                ;AUX SOUS-PROGRAMMES DE PAGINATION  CB.MAC

                                XLIST
                                LIST
                                TITLE   MONDAT  JOBDATA AREA EN PAGINATION
                                SUBTTL  REDEFINITION DE CERTAINES ADRESSES DE JOBDAT.REL
                                INTERN  USR41   ,USRAPR ,USRDDT ,USRENB ,USRFF  ,USRREL ,USRREN ,USRSA
                                INTERN  USRSYM  ,USRUSY ,USRDA

                                INTERN  JOBDA

                        000140  JORDA=140        ;POUR PROVOQUER MULTIPLY-DEFINED AU MOMENT D'UN CHARGEMENT


  000000'                       USR41:  BLOCK   1
  000001'                       USRAPR: BLOCK   1
  000002'                       USRDDT: BLOCK   1
  000003'                       USRENB: BLOCK   1
  000004'                       USRFF:  BLOCK   1
  000005'                       USRSYM: BLOCK   1
  000006'                       USRUSY: BLOCK   1
  000007'                       USRRFL: BLOCK   1
  000010'                       USRREN: BLOCK   1
  000011'                       USRSA:  BLOCK   1
  000012'                       USRDA:

                                        END

NO ERRORS DETECTED

PROGRAM BREAK IS 000012
```

```
MONDAT  JOBDATA AREA EN PAGINATION      MACRO.V43 10:48  2-SEP-70 PAGE 2
MONDAT.MAC      SYMBOL TABLE

ACU             000010  SPD     ADR             000004  SPD     ALAPRB          000003  SPD     ASCIIL          000001  SPD
BKSHFT  777777  777771  SPD     BLKSIZ          000200  SPD     CHN             000006  SPD     CORE            000011  SPD
DAT             000005  SPD     DDTOUT          000003  SPD     DEVCHR          000001  SPD     DRP             000000  SPD
DUMP            000017  SPD     EXIT            000012  SPD     HGHOFF          000002  SPD     HGHON           000003  SPD
HIGHSW  777777  777775  SPD     HORPAG          000004  SPD     IDX             000011  SPD     ILM             020000  SPD
ILMENB          000016  SPD     IMMB            000001  SPD     INS             000016  SPD     JDABLK          000001  SPD
JOBDA           000140  INT     JOBPAG          000140  SPD     LOALOC          400000  SPD     LOC             000007  SPD
MEMORY          000000  SPD     NADRB           000002  SPD     NORPAG          000010  SPD     OPTESS  777777  777777  SPD
PAGLOC          002000  SPD     PEEK            000033  SPD     PIL             000017  SPD     SD              000015  SPD
SETDDT          000002  SPD     SLEEP           000031  SPD     TAC             000001  SPD     TAC1            000002  SPD
TAC2            000003  SPD     TTY             000000  SPD     USR41           000008' INT     USRAPR          000001' INT
USRDA           000012' INT     USRDDT          000002' INT     USRENB          000003' INT     USRFF           000004' INT
USRREL          000007' INT     USRREN          000010' INT     USRSA           000011' INT     USRSYM          000005' INT
USRUSY          000006' INT     UUO             000016  SPD     UUOOFF          000001  SPD     UUOON           000000  SPD
UUOSW   777777  777775  SPD

2K CORE USED

ACU          5#
ADR          5#
ALAPRB       5#
ASCIIL       5#
BKSHFT       5#
BLKSIZ       5#
CHN          5#
CORE         5#
DAT          5#
DDTOUT       5#
DEVCHR       5#
DRP          5#
DUMP         5#
EXIT         5#
HGHOFF       5#
HGHON        5#
HIGHSW       5#
HORPAG       5#
IDX          5#
ILM          5#
ILMENB       5#
IMMB         5#      5
INS          5#
JDABLK       5#
JOBDA       11      13#
JOBPAG       5#
LOALOC       5#
LOC          5#
LSTCB        5
MEMORY       5#
NADRB        5#      5
NORPAG       5#
OPTESS       5
PAGLOC       5#
PEEK         5#
PIL          5#
SD           5#
SETDDT       5#
SLEEP        5#
TAC          5#      5
TAC1         5#
TAC2         5#
TTY          5#
USR41        8      16#
USRAPR       8      17#
USRDA        9      26#
USRDDT       8      18#
USRENB       8      19#
USRFF        8      20#
USRREL       8      23#
USRREN       8      24#
USRSA        8      25#
USRSYM       9      21#
USRUSY       9      22#
UUO          5#      5
UUOOFF       5#
UUOON        5#
UUOSW        5#      5
```

```
ANALYSE DE LA COMMANDE ET INITIALISATIONS DU MONITEUR  09,08,78 MACRO,V43 10:49  2-SEP-78 PAGE 1
CB     .MAC       DECLARATIONS

                                ;CETTE FILE A ETE ASSEMBLEE AVEC LA FILE DES PARAMETRES COMMUNS
                                ;AUX SOUS-PROGRAMMES DE PAGINATION  CB.MAC

                                XLIST
                                LIST
                                TITLE   ANALYSE DE LA COMMANDE ET INITIALISATIONS DU MONITEUR  09,08,78
                                SUBTTL  DECLARATIONS

  000000'                       WISEG
                                MLON

                                ;ASSEMBLAGE AVEC LA FILE CB.MAC POUR LES PARAMETRES GENERAUX ET POUR
                                ;COMPATIBILITE AVEC LES AUTRES SOUS-PROGRAMMES D'ALISSA

                                ;DEFINITIONS PARTICULIERES ET INTERNES
                                        ;DRAPEAUX NECESSAIRES A L'ANALYSE DE LA COMMANDE
                        000001  FINDRP==        1       ;RENCONTRE UN TERMINATEUR
                        000002  PJGDRP==        2       ;APRES UNE OUVERTURE DE CROCHETS DROITS
                        000004  NUMDRP==        4       ;MODE NUMERIQUE (PRJ,PRG OU NB DE K DE MEMOIRE)
                        000010  MESDRP==        10      ;MODE ALPHANUMERIQUE ORGANE,FILE OU EXTENSION
                        000020  PRJDRP==        20      ;NUMERO DE PROJET RENCONTRE
                        000040  ORGDRP==        40      ;ORGANE DEJA SPECIFIE (PERIPHERIQUE)
                        000100  FILDRP==        100     ;FILE DEJA SPECIFIEE
                        000200  EXTDRP==        200     ;EXTENSION DEJA SPECIFIEE
                        000400  PRGDRP==        400     ;NUMERO DE PROGRAMMEUR RENCONTRE

                                        ;ACCUMULATEURS
                        000002  CARACT==        2       ;TRANSFERT DE CARACTERE
                        000003  MOT==           3       ;ASEMBLAGE DU MOT ALPHANUMERIQUE
                        000004  NOMBRE==        4       ;ASSEMBLAGE D'UN NOMBRE
                        000005  MOTPTR==        5       ;POINTEUR D'ASSEMBLAGE DU MOT
                        000006  NMBPTR==        6       ;POINTEUR D'ASSEMBLAGE DU NOMBRE

                                        ;VARIABLES EXTERNES
                                EXTERN  PILPTR  ,ATEMPR ,ZTEMPR ,JOBHRL ,JOBSYM ,PAGELN
                                EXTERN  MEMBLK  ,MEMBLP ,MEMINI ,MEMGET ,MEMSYM ,PAGBLP ,ANYGET
                                EXTERN  MAXBLK  ,ABSMAX ,PAGADR ,USRREL ,CORUUO
                                EXTERN  PAGLNG  ,UTILNG ,ACCMEM ,METABS ,RNGMEM
                                EXTERN  JOB41   ,JOBDDT ,USRSYM ,USRUSY ,JOBSA  ,JOBREN ,JOBAPR ,PAGEPC
                                EXTERN  USR41   ,USRDDT ,USRSA  ,USRREN ,USRAPR ,USRFF  ,US1DDT ,USRDA
                                EXTERN  ARRSUP  ,TAPMES ,USERGO ,US1APO
                                EXTERN  TTYHHI  ,MODUMP ,MEMFIL

                                        ;VARIABLES INTERNES
                                INTERN  DEPART


                                SUBTTL  DEPART ET ANALYSE DE LA LIGNE

                                ;DEBUT DE TRAITEMENT D'UN PROGRAMME PAGINE = IL FAUT D'ABORD ANALYSER
                                ;LA COMMANDE DE L'UTILISATEUR APRES AVERTISSEMENT AU MOYEN DE L'ASTERISUE

  000000' 047000  000000  DEPART: CALLI   0       ;REMISE A ZERO DE TOUS LES CANAUX ET CONVENTIONS
                                                  ;PREALABLES AVEC LE MONITEUR
  000001' 201040  000000          MOVEI   TAC,0           ;POUR DEMANDER L'ECRITURE
  000002' 047040  000036          CALLI   TAC,<SETUWP==36>;PERMETTRE L'ECRITURE EN HAUT
  000003' 254200  000004'         HALT    .+1             ;TANT PIS = C'EST SURTOUT POUR LE DDT
  000004' 200740  000000*         MOVE    PIL,PILPTR      ;INITIALISER LA PILE DE TRAVAIL
  000005' 403000  000000*         SETZB   DRP,ATEMPR      ;PREMIERE POSITION TEMPORAIRE A REMETTRE A ZERO
  000006' 200040  000377'         MOVE    TAC,[XWD ATEMPR,ATEMPR+1]       ;POINTEUR DE BLT
  000007' 251040  000000*         BLT     TAC,ZTEMPR      ;REMISE A ZERO DE A JUSQU'A Z
                                  IFN     OPTESSAIS <
  000010' 260740  000333' SMPTCH: PUSHJ   PIL,BLTSYM >    ;UNE SEULE FOIS (VOIR BLTSYM)
  000011' 041000  000001          INIT    TTY,ASCIIL      ;INITIALISER LE TELETYPE
  000012' 646471  000000          SIXBIT  .TTY.
  000013' 000000  000000*         XWD     0,TTYHHI        ;ADRESSE DE LA TETE DES TETES DE TAMPONS
  000014' 047000  000012          CALLI   EXIT            ;NE DEVRAIT PAS ARRIVER
  000015' 201040  000400'         MOVEI   TAC,[ASCIZ .*.] ;SORTIR L'ASTERISTIQUE
  000016' 047040  000003          CALLI   TAC,DDTOUT      ;EN MODE DDT
  000017' 064000  000001          INBUF   TTY,1           ;UN SEUL BUFFER D'ENTREE
  000020' 066000  000000          INPUT   TTY,0           ;ENTRER LA LIGNE DE COMMANDE
  000021' 201140  000000  MSUITE: MOVEI   MOT,0           ;SUITE POUR UN NOUVEAU MOT
  000022' 200240  000401'         MOVE    MOTPTR,[POINT 6,MOT]
  000023' 201200  000000  NSUITE: MOVEI   NOMBRE,0        ;SUITE POUR UN NOMBRE EVENTUEL
  000024' 200300  000402'         MOVE    NMBPTR,[POINT 3,NOMBRE,17]
  000025' 375000  000002* SUITE:  SOSGE   TTYHHI+2        ;CONTROLE DU VIDAGE DU TAMPON
  000026' 254000  000175'         JRST    CMDEND          ;FIN DE LA COMMANDE
  000027' 134100  000001*         ILDB    CARACT,TTYHHI+1 ;CARACTERE SUIVANT DANS LE TAMPON
  000030' 623000  000001          TLZE    DRP,FINDRP      ;DEJA RENCONTRE UN TERMINATEUR ?
  000031' 254000  000175'         JRST    CMDEND          ;OUI = FIN DE LA COMMANDE


                                ;DEBUT DE RECONNAISSANCE DU GENRE DE CARACTERE

  000032' 307100  000132          CAIG    CARACT,"Z"      ;EST-CE ALPHABETIQUE ?
  000033' 305100  000101          CAIGE   CARACT,"A"      ;...
  000034' 254000  000045'         JRST    NALPHA          ;NON = NON-ALPHABETIQUE
  000035' 607000  000002          TLNN    DRP,PJGDRP      ;OUI = EST-CE EN COURS DE COUPLE PJ,PG ?
  000036' 603000  000004          TLNE    DRP,NUMDRP      ;NON = OU D'UN NOMBRE DE K ?
  000037' 254000  000166'         JRST    ILLEGA          ;OUI = COMMANDE ILLEGALE
  000040' 661000  000010  ALPHNU: TLO     DRP,MESDRP      ;MARQUER QU'UN MOT OU MESSAGE EST ASSEMBLE
  000041' 275100  000040          SUBI    CARACT,40       ;CONVERSION EN SIXBIT
  000042' 603240  770000          TLNE    MOTPTR,770000   ;DEJA 6 CARACTERES ?
  000043' 136100  000005          IDPB    CARACT,MOTPTR   ;NON = ASSEMBLER
  000044' 254000  000025'         JRST    SUITE           ;ALLER CHERCHER LE CARACTERE SUIVANT


                                ;CARACTERE NON ALPHABETIQUE

  000045' 307100  000071  NALPHA: CAIG    CARACT,"9"      ;EST-CE UN CHIFFRE ?
  000046' 305100  000060          CAIGE   CARACT,"0"      ;...
  000047' 254000  000067'         JRST    NONUME          ;NON = NON-NUMERIQUE
  000050' 623000  000120          TLNE    DRP,PRJDRP!FILDRP       ;DEJA LA FILE OU [PJ,PG]
  000051' 623000  000010          TLNE    DRP,MESDRP      ;OUI = ALORS SUREMENT PAS DE MOT
  000052' 254000  000040'         JRST    ALPHNU          ;NON = PRENDRE COMME UN MOT (FILE ,EXT...)
  000053' 607000  000002          TLNN    DRP,PJGDRP      ;EN COURS DE COUPLE [PJ,PG] ?
  000054' 254000  000063'         JRST    DECIMA          ;NON = ALORS QUANTITE DECIMALE
  000055' 301100  000070          CAIL    CARACT,"8"      ;EST-CE BIEN OCTAL ?
  000056' 254000  000166'         JRST    ILLEGA          ;NON = COMMANDE ILLEGALE
  000057' 661000  000004          TLO     DRP,NUMDRP      ;OUI = MARQUER QUE C'EST NUMERIQUE
  000060' 603300  770000          TLNE    NMBPTR,770000   ;DEJA 6 CHIFFRES ?
  000061' 136100  000006          IDPB    CARACT,NMBPTR   ;NON = ASSEMBLER
  000062' 254000  000025'         JRST    SUITE           ;EN TOUS CAS ALLER CHERCHER LE SUIVANT
  000063' 661000  000004  DECIMA: TLO     DRP,NUMDRP      ;PRECISER QUE C'EST NUMERIQUE
  000064' 221200  000012          IMULI   NOMBRE,+D10     ;DECALER LE RESULTAT ACCUMULE AU PREALABLE
  000065' 271200  777720          ADDI    NOMBRE,-"0"(CARACT)     ;ET AJOUTER CE DERNIER CHIFFRE
  000066' 254000  000025'         JRST    SUITE           ;PUIS ALLER CHERCHER LE CARACTERE SUIVANT
```

```
ANALYSE DE LA COMMANDE ET INITIALISATIONS DU MONITEUR  09,08.70 MACRO.V43 10:49  2-SEP-70 PAGE 2
ANALYS.MAC       DEPART ET ANALYSE DE LA LIGNE

000067' 302100  000072  NONUME: CAIE    CARACT,"I"        ;SEPARATEUR "I" ?
000070' 254000  000077'         JRST    N2POIN            ;NON - CHERCHER SI C'EST "."
000071' 627000  000010          TLZN    DRP,MESDRP        ;OUI - Y-A-T IL EU MOT ASSEMBLE ?
000072' 205140  446353          MOVSI   MOT,(SIXBIT .DSK.);NON - SUPPOSER QUE C'EST LE DISQUE
000073' 663000  000040          TLOE    DRP,ORGDRP        ;DEJA EU "I" ?
000074' 254000  000166'         JRST    ILLEGA            ;OUI - COMMANDE ILLEGALE
000075' 202140  000001*         MOVEM   MOT,MODUMP+1      ;NON - PREMIERE ET UNIQUE PRECISION DU
000076' 254000  000021'         JRST    MSUITE            ;PERIPHERIQUE SERVANT DE MEMOIRE

000077' 302100  000056  N2POIN: CAIE    CARACT,"."        ;SEPARATEUR "." ?
000100' 254000  000106'         JRST    N1POIN            ;NON - CHERCHER SI C'EST "["
000101' 603000  000010          TLNE    DRP,MESDRP        ;OUI - UN MOT AU MOINS A ETE ASSEMBLE ?
000102' 663000  000100          TLOE    DRP,FILDRP        ;OUI - PREMIERE ET UNIQUE PRECISION DU
                                                          ;NOM DE FILE ?
000103' 254000  000166'         JRST    ILLEGA            ;NON - COMMANDE ILLEGALE
000104' 202140  000000* CEFILE: MOVEM   MOT,MEMFIL         ;OUI - RANGER LE NOM DE LA FILE SERVANT
000105' 254000  000021'         JRST    MSUITE            ;D'EXTENSION MEMOIRE

                                ,RECONNAISSANCE DES CARACTERES SEPARATEURS (SUITE)

000106' 302100  000133  N1POIN: CAIE    CARACT,"["        ;SEPARATEUR "[" ?
000107' 254000  000123'         JRST    NONOUV            ;NON - CHERCHER SI C'EST "."
000110' 667000  000022          TLON    DRP,PJGDRP!PRJDRP;OUI - DEJA EU "[" OU [XXX,YYY] ?
000111' 254000  000166'         JRST    ILLEGA            ;OUI - COMMANDE ILLEGALE
000112' 621000  000020          TLZ     DRP,PRJDRP        ;PAIRE PRJ,PRG NON ENCORE RECUE
000113' 627000  000010          TLZN    DRP,MESDRP        ;Y-A-T-IL UN MESSAGE A RANGER ?
000114' 254000  000025'         JRST    SUITE             ;NON - C'EST SANS DOUTE DEJA FAIT
000115' 667000  000100  CEXTEN: TLON    DRP,FILDRP        ;DEJA EU LA FILE ?
000116' 254000  000104'         JRST    CEFILE            ;NON - CE DOIT L'ETRE
000117' 663000  000200          TLOE    DRP,EXTDRP        ;DEJA EUE L'EXTENSION ?
000120' 254000  000166'         JRST    ILLEGA            ;OUI - COMMANDE ILLEGALE
000121' 512140  000001*         HLLZM   MOT,MEMFIL+1      ;NON - RANGER L'EXTENSION
000122' 254000  000025'         JRST    SUITE             ;ET ALLER CHERCHER LE CARACTERE SUIVANT
000123' 302100  000054  NONOUV: CAIE    CARACT,","        ;SEPARATEUR "," ?
000124' 254000  000142'         JRST    NCOMMA            ;NON - CHERCHER SI C'EST "]"
000125' 603000  000002          TLNE    DRP,PJGDRP        ;OUI - DANS [XXX,YYY] ?
000126' 663000  000400          TLOE    DRP,PRGDRP        ;OUI - DEJA EU [XXX, ?
000127' 254000  000166'         JRST    ILLEGA            ;NON - OUI - COMMANDE ILLEGALE
000130' 627000  000004          TLZN    DRP,NUMDRP        ;      NON - OUI - COMMANDE ILLEGALE
000131' 254000  000166'         JRST    ILLEGA            ;NON - COMMANDE ILLEGALE
000132' 260740  000135' CESPRJ: PUSHJ   PIL,DROITE        ;OUI - CADRER A DROITE
000133' 506200  000003*         HRLM    NOMBRE,MEMFIL+3   ;ET RANGER EN BONNE PLACE
000134' 254000  000023'         JRST    NSUITE            ;CONTINUER AVEC DES CHIFFRES

000135' 607300  770000  DROITE: TLNN    NMBPTR,770000     ;CADRE A DROITE ?
000136' 263740  000000          POPJ    PIL,0             ;OUI = RETOUR
000137' 133300  000000          IBP     NMBPTR,0          ;INCREMENTER LE POINTEUR DE BYTES
000140' 242200  777775          LSH     NOMBRE,=3         ;ET DECALER DE 1 CHIFFRE OCTAL
000141' 254000  000135'         JRST    DROITE            ;AVANT DE BOUCLER

000142' 302100  000135  NCOMMA: CAIE    CARACT,"]"        ;SEPARATEUR "]" ?
000143' 254000  000155'         JRST    NOFERM            ;NON = CHERCHER SI C'EST ESPACE OU TABULATION
000144' 623000  000002          TLZE    DRP,PJGDRP        ;OUI = Y=A=T=IL EU "[" ?
000145' 627000  000400          TLZN    DRP,PRGDRP        ;OUI = ET XXX, ?
000146' 254000  000166'         JRST    ILLEGA            ;NON = COMMANDE ILLEGALE
000147' 627000  000004  NOCORE: TLZN    DRP,NUMDRP        ;Y=A=T=IL EU AU MOINS UN CHIFFRE ?
000150' 254000  000166'         JRST    ILLEGA            ;NON = ALORS COMMANDE ILLEGALE
000151' 661000  000020  CESPRG: TLO     DRP,PRJDRP        ;MARQUER QUE [XXX,YYY] A ETE RECU
000152' 260740  000135'         PUSHJ   PIL,DROITE        ;CADRAGE DU NUMERO DE PROGRAMMEUR
000153' 542200  000003*         HRRM    NOMBRE,MEMFIL+3   ;ET RANGER EN BONNE PLACE
000154' 254000  000023'         JRST    NSUITE            ;POURSUIVRE AVEC LE CARACTERE SUIVANT
```

```
                                .RECONNAISSANCE DES CARACTERES SEPARATEURS (SUITE)

000155' 302100  000040  NOFERM: CAIE    CARACT," "        ;EST-CE UN ESPACE ?
000156' 306100  000011          CAIN    CARACT,"          ";OU UNE TABULATION ?
                                JRST    [TLZE   ORP,MESORP        ;LE MOT A T-IL ETE ASSEMBLE ?
                                         JRST    CEXTEN           ;OUI - VOIR SI C'EST FILE OU EXT
                                         TLZN    ORP,NUMORP       ;UN NOMBRE A-T-IL ETE ASSEMBLE
                                         JRST    SUITE            ;NON - ESPACES DE TETE
                                         TLNN    ORP,PJGORP       ;EN COURS DE [XXX , YYY] ?
                                         JRST    CMDEN0           ;NON - CE DOIT ETRE LE NOMBRE DE K
                                         TLNN    ORP,PRGORP       ;APRES NUMERO DE PROGRAMMEUR ?
                                         JRST    CESPRJ           ;NON-O - APRES PRJ
000157' 254000  000403'                  JRST    CESPRG]          ;OUI - RANGER LE NUMERO DE PROJET
000160' 302100  000015          CAIE    CARACT,15        ;SEPARATEUR EST <RET CHARIOT> ?
000161' 306100  000012          CAIN    CARACT,12        ;OU <LINE FEED> ?
                                JRST    [TLOE   ORP,FINORP        ;DEJA EU UN TERMINATEUR ?
                                         JRST    CMDEND           ;OUI = COMMANDE TERMINEE
                                         TLZE    ORP,PJGORP       ;NON = EU PROJET PROGRAMMEUR ?
                                         JRST    NOCORE           ;OUI = RANGER ARGUMENT EVENTUEL
                                         TLZE    ORP,MESORP       ;UN MOT ASSEMBLE ?
                                         JRST    CEXTEN           ;OUI = VOIR LEQUEL
                                         TLZE    ORP,NUMORP       ;UN NOMBRE ASSEMBLE
                                         JRST    CMDEN0           ;OUI
000162' 254000  000414'                  JRST    CMDEND]          ;NON = FIN DE LA COMMANDE
000163' 302100  000175          CAIE    CARACT,175       ;SEPARATEUR EST ALTMODE ?
000164' 306100  000013          CAIN    CARACT,13        ;OU VERTICAL=TAB ?
000165' 254020  000162'         JRST    .+3              ;OUI = COMME CR OU LF

000166' 201000  000362' ILLEGA: MOVEI   CMDMES            ;ADRESSE DU MESSAGE
000167' 047000  000003          CALLI   DDTOUT           ;SORTIE EN DDT
000170' 254000  000000'         JRST    DEPART           ;RECOMMENCER A ZERO
000171' 322200  000175' CMDEN0: JUMPE   NOMBRE,CMDEND    ;PAS DE NOMBRE DE K
000172' 221200  002000          IMULI   NOMBRE,2000      ;CONVERSION EN MOTS
000173' 275200  000001          SUBI    NOMBRE,1         ;ADRESSE=1
000174' 202200  000000*         MOVEM   NOMBRE,ARSMAX    ;RANGER

                                SUBTTL  INITIALISATIONS PREALABLES AU LANCEMENT DE LA PAGE DE DEBUT

000175' 071000  000000  CMDEND: RELEASE TTY,0            ;FIN DE L'UTILISATION DE TTY
000176' 205040  446353          MOVSI   TAC,(SIXBIT .DSK.);STANDARD
000177' 336000  000001*         SKIPN   MODUMP+1         ;ORGANE PRECISE ?
000200' 202040  000001*         MOVEM   TAC,MODUMP+1     ;NON ALORS ESSAYER AVEC LE DISQUE
000201' 201040  000017          MOVEI   TAC,DUMP         ;MODE NORMAL
000202' 202040  000000*         MOVEM   TAC,MODUMP       ;INITIALISER
000203' 205040  635766          MOVSI   TAC,(SIXBIT .SOV.);EXTENSION STANDARD
000204' 336000  000001*         SKIPN   MEMFIL+1         ;EXTENSION PRECISEE ?
000205' 202040  000001*         MOVEM   TAC,MEMFIL+1     ;NON ALORS ESSAYER LA VALEUR STANDARD
000206' 260740  000000*         PUSHJ   PIL,MEMINI       ;INITIALISER LE CANAL "MEMORY"
000207' 200040  000425'         MOVE    TAC,[IOWD BLKSIZ,MEMBLK];PREPARER L'IOWD PERMANENT POUR LE
000210' 202040  000000*         MOVEM   TAC,MEMBUP       ;BLOC MEMOIRE TEMPORAIRE
000211' 201100  777777          MOVEI   TAC1,-JOABLK     ;POUR LE PREMIER BLOC DE LA FILE MEMOIRE
000212' 260740  000000*         PUSHJ   PIL,MEMGET       ;ENTRER LE PREMIER BLOC DE JOB-DATA-AREA
```

```
000213' 336040  000140*          SKIPN   TAC,JOBPAG+MEMBLK;PAGINE CONVENABLEMENT AU MOINS ?
000214' 254000  000356'          JRST    FMTERR          ;NON ERREUR DE FORMAT
000215' 275040  000001           SUBI    TAC,1           ;S'ASSURER QUE LA PAGE FAIT BIEN UN NB ENTIER DE K
000216' 405040  776000           ANDI    TAC,776000
000217' 271040  002000           ADDI    TAC,2000
000220' 202040  000000*          MOVEM   TAC,PAGLNG      ;ET RANGER DEFINITIVEMENT
000221' 201101  400000           MOVEI   TAC1,400000(TAC);ADRESSE DE DEBUT DE LA PAGE 1
000222' 202100  000000*          MOVEM   TAC1,PAG0LN     ;COMME LONGUEUR DE LA PAGE 0
000223' 554100  000140*          HLRZ    TAC1,JOBPAG+MEMBLK;IDEM POUR LA BASE DE LA PAGE
000224' 271040  001777           ADDI    TAC,PAGLOC-1    ;DANS LE SEGMENT BAS
000225' 047040  000011           CALLI   TAC,CORE        ;ALLOCATION DE MEMOIRE REELLE JUSTE SUFFISANTE
000226' 254000  000360'          JRST    ADJERR          ;ERREUR DANS L'AJUSTEMENT DE LA MEMOIRE
                                 DEFINE  X(A),<IRP A
                                 <EXTERN MEM'A
                                 MOVE    TAC,MEM'A
                                 MOVEM   TAC,USR'A>>
                                 X       <41,DDT,APR,REN,FF,SA>+IRP
000227' 200040  000000*          MOVE    TAC,MEM'41
000230' 202040  000000*          MOVEM   TAC,USR'41
000231' 200040  000000*          MOVE    TAC,MEM'DDT
000232' 202040  000000*          MOVEM   TAC,USR'DDT
000233' 200040  000000*          MOVE    TAC,MEM'APR
000234' 202040  000000*          MOVEM   TAC,USR'APR
000235' 200040  000000*          MOVE    TAC,MEM'REN
000236' 202040  000000*          MOVEM   TAC,USR'REN
000237' 200040  000000*          MOVE    TAC,MEM'FF
000240' 202040  000000*          MOVEM   TAC,USR'FF
000241' 200040  000000*          MOVE    TAC,MEM'SA
000242' 202040  000000*          MOVEM   TAC,USR'SA
000243' 552040  000000*          HRRZM   TAC,PAGEPC      ;ADRESSE DE DEPART
000244' 505040  264000           HRLI    TAC,(JSR)       ;JSR ...
000245' 502040  000000*          HLLM    TAC,JOB41       ;SUPPRIMER LE HALT DE CHARGEMENT
000246' 201040  000000*          MOVEI   TAC,US1DDT
000247' 332000  000232*          SKIPE   USRDDT          ;SEULEMENT SI LE DDT
000250' 047040  000002           CALLI   TAC,SETDDT      ;PLACER L'ADRESSE DE RENVOI AU DDT EN ZONE PROTEGEE
000251' 200040  000426'          MOVE    TAC,[XWD MEMSYM,USRSYM]
000252' 251040  000000*          BLT     TAC,USRUSY      ;POUR LA TABLE DES SYMBOLES
000253' 550040  000000*          HRRZ    TAC,USRSYM      ;ADRESSE LOCALE
000254' 275040  000200           SUBI    TAC,JOBBLK*BLKSIZE;- CE QUI N'EXISTE PLUS
000255' 270040  000222*          ADD     TAC,PAG0LN      ;+BASE + PAGE 0
000256' 542040  000253*          HRRM    TAC,USRSYM      ;A JOUR

                                ;PREPARER L'IOWD DE PAGE

000257' 210040  000000*          MOVN    TAC,UTILNG      ;-LONGUEUR DE PAGE
000260' 505040  002000           HRLI    TAC,PAGLOC      ;ADRESSE DE DEPART DE LA PAGE
000261' 207000  000001           MOVSS   TAC             ;COMME IOWD
000262' 275040  000001           SUBI    TAC,1           ;...POINTER SOUS LE DEBUT DE PAGE
000263' 202040  000000*          MOVEM   TAC,PAGBLP      ;RANGER EN DUE PLACE
000264' 402000  000001*          SETZM   PAGBLP+1        ;ASSURER UN VERITABLE ET UNIQUE IOWD
                                ;DETERMINATION EXACTE DU HAUT DE LA MEMOIRE AU DEPART

000265' 554040  000241*          HLRZ    TAC,MEMSA       ;JOBFF
000266' 552040  000240*          HRRZM   TAC,USRFF       ;GARDER LA VALEUR ABSOLUE
000267' 260740  000000*          PUSHJ   PIL,ARRSUP      ;ARRONDIR A LA PAGE SUPERIEURE +1
000270' 261740  000001           PUSH    PIL,TAC         ;AVANT CORUUO
000271' 260740  000000*          PUSHJ   PIL,CORUUO      ;AJUSTAGE DE LA MEMOIRE
000272' 262740  000000*          POP     PIL,USRRFL      ;MISE A JOUR ASSUREE
000273' 200040  000257*          MOVE    TAC,UTILNG      ;POUR CONVERTIR EN NOMBRE DE BLOCS
000274' 242040  777771           LSH     TAC,BKSHFT      ;LE NOMBRE DE BLOCS UTILISES PAR PAGE
000275' 221101  000000           IMULI   TAC1,(TAC)
000276' 201040  000263*          MOVEI   TAC,PAGBLP      ;LIRE DANS LA ZONE PAGE
000277' 260740  000000*          PUSHJ   PIL,ANYGET      ;(TAC1)=NUMERO DE BLOC QUE L'ON DESIREREAIT
000300' 344100  000302'          AOJA    TAC1,.+2        ;TROUVE SANS EOF
000301' 367100  000277'          SOJG    TAC1,.-2        ;FIN DE FILE - CHERCHER LE PRECEDENT
000302' 260740  000277*          PUSHJ   PIL,ANYGET      ;ESSAYER EN CROISSANT MAINTENANT
000303' 344100  000302'          AOJA    TAC1,.-1        ;PAS ENCORE DE FIN DE FILE
000304' 275100  000001           SUBI    TAC1,1          ;DERNIER BLOC DE LA MEMOIRE
000305' 202100  000000*          MOVEM   TAC1,MAXBLK     ;RANGER POUR UTILISATION EN CAS D'EXTENSION
000306' 200040  000427'          MOVE    TAC,[XWD MEMBLK,PAGLOC];POINTEUR DE BLT
000307' 251040  002177           BLT     TAC,PAGLOC+BLKSIZE-1;BLT DE JOBDATA AREA EN PAGE
000310' 402000  002200           SETZM   PAGLOC+BLKSIZE  ;0 = TERMINATEUR DES COMMANDES
000311' 201440  002000*          MOVEI   IDX,PAGLOC+USRDA;DEBUT DES COMMANDES
000312' 550211  000000  PREABS:  HRRZ    ADR,(IDX)       ;COMMANDE SUIVANTE
000313' 322200  000330'          JUMPE   ADR,PREAB1      ;SI 0 C'EST LA FIN
000314' 315200  000255*          CAMGE   ADR,PAG0LN      ;ADRESSE ABSOLUE
000315' 270200  000220*          ADD     ADR,PAGLNG      ;NON - CE DOIT ETRE PAGE 1
000316' 260740  000000*          PUSHJ   PIL,ACCMEM      ;ALLER LIRE CE MOT POINTE PAR ADR
000317' 556340  000000*          HLRZM   LOC,PAGADR      ;POUR MISE EN ABSOLU DE
000320' 200140  000004           MOVE    TAC2,ADR        ;SAUVER ADR POUR RNGMEM
000321' 550207  000000           HRRZ    ADR,(LOC)       ;ADR=CE QUI EST LOCALEMENT EN (LOC)
000322' 260740  000000*          PUSHJ   PIL,METABS      ;MISE EN ABSOLU EVENTUELLE
000323' 542207  000000           HRRM    ADR,(LOC)       ;RANGEMENT...
000324' 402000  000317*          SETZM   PAGADR          ;PRECAUTION POUR LE RANGEMENT ET L'ACCES
000325' 200200  000003           MOVE    ADR,TAC2        ;RECUPERER ADR
000326' 260740  000000*          PUSHJ   PIL,RNGMEM      ;REECRITURE SUR LE DISQUE
000327' 344440  000312'          AOJA    IDX,PREABS      ;AU SUIVANT...
000330' 550200  000242* PREAB1:  HRRZ    ADR,USRSA       ;ADRESSE ABSOLUE DE DEPART
000331' 202200  000243*          MOVEM   ADR,PAGEPC      ;AVEC 0 COMME DRAPEAUX
000332' 254000  000000*          JRST    US1AP0          ;DE MEMOIRE- LANCER LE PROGRAMME MAINTENANT

                                IFN     OPTESSAIS,<
                                ;BLT DE LA TABLE DES SYMBOLES

000333' 574040  000000* BLTSYM:  HLRE    TAC,JOBSYM      ;NMB DE SYMBOLES * 2
000334' 213000  000001           MOVNS   TAC             ;+...
000335' 554100  000000*          HLRZ    TAC1,JOBHRL     ;FIRST FREE IN HISEG
000336' 271101  400000           ADDI    TAC1,400000(TAC);DERNIERE ADRESSE REQUISE
000337' 550040  000335* BLTSM1:  HRRZ    TAC,JOBHRL      ;HIGHEST IN HI-SEG
000340' 307101  000000           CAIG    TAC1,(TAC)      ;ASSEZ ?
000341' 254000  000346'          JRST    BLTSM2          ;OUI..
000342' 515042  000000           HRLZI   TAC,(TAC1);POUR RECLAMER DANS LE HAUT SEULEMENT
000343' 047040  000011           CALLI   TAC,CORE
000344' 254000  000360'          JRST    ADJERR          ;PAS POSSIBLE D'AVOIR CE QUI EST DEMANDE
000345' 254000  000337'          JRST    BLTSM1          ;ACCORDE...
000346' 554040  000337* BLTSM2:  HLRZ    TAC,JOBHRL      ;FIRST FREE
000347' 271040  400000           ADDI    TAC,400000      ;DEBUT DU HAUT SEGMENT
000350' 504040  000333*          HRL     TAC,JOBSYM      ;ADRESSE SOURCE
000351' 542040  000350*          HRRM    TAC,JOBSYM      ;FUTURE ADRESSE DE LA TABLE DES SYMBOLES
000352' 251042  000000           BLT     TAC,(TAC1)      ;JUSQU'A LA LIMITE
000353' 205040  255000           MOVSI   TAC,(JFCL)      ;POUR NE PAS RECOMMENCER
000354' 202040  000010'          MOVEM   TAC,SMPTCH      ;PATCH DU PUSHJ D'APPEL
000355' 263740  000000           POPJ    PIL,0           ;EXIT
```

```
ANALYSE DE LA COMMANDE ET INITIALISATIONS DU MONITEUR  09.08.70 MACRO.V43 10:49  2-SEP-70 PAGE 5
ANALYS.MAC      INITIALISATIONS PREALABLES AU LANCEMENT DE LA PAGE DE DEBUT

                                .ROUTINE POUR ERREUR DE FORMAT

000356' 201000  000366' FMTERR: MOVEI   FMTMES          ;ADRESSE DU MESSAGE
000357' 254000  000000*         JRST    TAPMES

000360' 201000  000373' ADJERR: MOVEI   ADJMES          ;ADRESSE DU MESSAGE
000361' 254000  000357*         JRST    TAPMES

000362' 416371  546602  CMDMES: ASCIZ   .COMMANDE ILLEGALE.
        000363' 472110  520222
        000364' 462310  543602
        000365' 462120  000000
000366' 466032  553202  FMTMES: ASCIZ   .MAUVAIS FORMAT D'ENTREE.
        000367' 446464  043236
        000370' 512330  152100
        000371' 421170  547250
        000372' 512130  500000
000373' 502030  742500  ADJMES: ASCIZ   .PAGE TROP LONGUE.
        000374' 522451  750100
        000375' 462371  643652
        000376' 424000  000000
000377'                 LIT
000377' 000000* 000201*
000400' 250000  000000
000401' 440600  000003
000402' 220300  000004
000403' 623000  000010
000404' 254000  000115'
000405' 627000  000004
000406' 254000  000025'
000407' 607000  000002
000410' 254000  000171'
000411' 607000  000400
000412' 254000  000132'
000413' 254000  000151'
000414' 663000  000001
000415' 254000  000175'
000416' 623000  000002
000417' 254000  000147'
000420' 623000  000010
000421' 254000  000115'
000422' 623000  000004
000423' 254000  000171'
000424' 254000  000175'
000425' 777600  777777*
000426' 000000* 000256*
000427' 000000* 002000
000430'                 VAR
                        END     JRST    DEPART          ;DEBUT DE L'EXECUTION EN PAGINATION

NO ERRORS DETECTED

PROGRAM BREAK IS 000430
```

```
ANALYSE DE LA COMMANDE ET INITIALISATIONS DU MONITEUR  09.08.78 MACRO.V43 10:49  2-SEP-78 PAGE 6
ANALYS.MAC      SYMBOL TABLE

ABSMAX          000174' EXT     ACCMEM          000316' EXT     ACU             000010  SPD     ADJERR          000360'
ADJMES          000373'         ADR             000004  SPD     ALAPRB          000003  SPD     ALPHNU          000040'
ANYGET          000302' EXT     ARRSUP          000267' EXT     ASCIIL          000001  SPD     ATEMPR          000005' EXT
BKSHFT  777777  777771  SPD     BLKSIZ          000200  SPD     BLTSM1          000337'         BLTSM2          000346'
BLTSYM          000333'         CARACT          000002  SPD     CEFILE          000104'         CESPRG          000151'
CESPRJ          000132'         CEXTEN          000115'         CHN             000006  SPD     CMDEND          000171'
CMDEND          000175'         CMDMES          000362'         CORE            000011  SPD     CORUUO          000271' EXT
DAT             000005  SPD     DDTOUT          000003  SPD     DECIMA          000063'         DEPART          000000' INT
DEVCHR          000001  SPD     DROITE          000135'         DRP             000000  SPD     DUMP            000017  SPD
EXIT            000012  SPD     EXTDRP          000200  SPD     FILDRP          000100  SPD     FINDRP          000001  SPD
FMTERR          000356'         FMTMES          000366'         HGHOFF          000002  SPD     HGHON           000003  SPD
HIGHSW  777777  777775  SPD     HORPAG          000004  SPD     IDX             000011  SPD     ILLEGA          000166'
ILM             020000  SPD     ILMENB          000016  SPD     IMMB            000001  SPD     INS             000016  SPD
JDABLK          000001  SPD     JOB41           000245' EXT     JOBAPR          000000  EXT     JOBDDT          000000  EXT
JOBHRL          000346' EXT     JOBPAG          000140  SPD     JOBREN          000000  EXT     JOBSA           000000  EXT
JOBSYM          000351' EXT     LOALOC          400000  SPD     LOC             000007  SPD     MAXBLK          000305' EXT
MEM41           000227' EXT     MEMAPR          000233' EXT     MEMBLK          000000  EXT     MEMBLP          000210' EXT
MEMDDT          000231' EXT     MEMFF           000237' EXT     MEMFIL          000104' EXT     MEMGET          000212' EXT
MEMINI          000206' EXT     MEMORY          000000  SPD     MEMREN          000235' EXT     MEMSA           000265' EXT
MEMSYM          000000  EXT     MESDRP          000010  SPD     METABS          000322' EXT     MODUMP          000202' EXT
MOT             000003  SPD     MOTPTR          000005  SPD     MSUITE          000021'         N1POIN          000106'
N2POIN          000077'         NADRB           000002  SPD     NALPHA          000045'         NCOMMA          000142'
NMBPTR          000006  SPD     NOCORE          000147'         NOFERM          000155'         NOMBRE          000004  SPD
NONOUV          000123'         NONUME          000067'         NORPAG          000010  SPD     NSUITE          000023'
NUMDRP          000004  SPD     OPTESS  777777  777777  SPD     ORGDRP          000040  SPD     PAGBLN          000314' EXT
PAGADR          000324' EXT     PAGBLP          000276' EXT     PAGEPC          000331' EXT     PAGLNG          000315' EXT
PAGLOC          002000  SPD     PEEK            000033  SPD     PIL             000017  SPD     PILPTR          000004' EXT
PJGDRP          000002  SPD     PREAB1          000330'         PREABS          000312'         PRGDRP          000400  SPD
PRJDRP          000020  SPD     RNGMEM          000326' EXT     SD              000015  SPD     SETDDT          000002  SPD
SETUWP          000036  SPD     SLEEP           000031  SPD     SMPTCH          000010'         SUITE           000025'
TAC             000001  SPD     TAC1            000002  SPD     TAC2            000003  SPD     TAPMES          000361' EXT
TTY             000000  SPD     TTYHWI          000013' EXT     US1APB          000332' EXT     US1DDT          000246' EXT
USERGO          000000  EXT     USR41           000230' EXT     USRAPR          000234' EXT     USRDA           000000  EXT
USRDDT          000247' EXT     USRFF           000266' EXT     USRREL          000272' EXT     USRREN          000236' EXT
USRSA           000330' EXT     USRSYM          000426' EXT     USRUSY          000252' EXT     UTILNG          000273' EXT
UUO             000016  SPD     UUOOFF          000001  SPD     UUOON           000000  SPD     UUOSW   777777  777775  SPD
ZTEMPR          000007' EXT

3K CORE USED
```

ABSMAX	37#	211												
ACCMEM	38#	306												
ACU	5#													
ADJERR	240	332	348#											
ADJMES	348	353#												
ADR	5#	302	303	304	305	308	309	311	313	316	317			
ALAPRB	5#													
ALPHNU	90#	103												
ANYGET	36#	291	294											
ARRSUP	41#	283												
ASCIIL	5#	64												
ATEMPR	35#	59	60											
BKSHFT	5#	288												
BLKSIZ	5#	225	267	299	300									
BLTSM1	327#	333												
BLTSM2	329	334#												
BLTSYM	63	323#												
CARACT	28#	78	84	85	91	93	98	99	106	110	114	116	125	136
	149	166	179	180	190	191	201	202						
CEFILE	131#	144												
CESPRG	173#	189												
CESPRJ	156#	188												
CEXTEN	143#	182	197											
CHN	5#													
CMDEND	186	199	208#											
CMDEND	77	80	193	200	208	215#								
CMDMES	205	351#												
CORE	5#	239	331											
CORUUB	37#	285												
DAT	5#													
DDTOUT	5#	69	206											
DECIMA	105	112#												
DEPART	45	53#	207	301										
DEVCHR	5#													
DROITE	156	160#	164	174										
DRP	5#	59	79	87	88	98	101	102	104	108	112	118	120	127
	128	138	140	141	143	145	151	152	154	168	169	171	173	181
	183	185	187	192	194	196	198							
DUMP	5#	219												
EXIT	5#	67												
EXTDRP	24#	145												
FILDRP	23#	101	128	143										
FINDRP	17#	79	192											
FMTERR	230	345#												
FMTMES	345	352#												
HGHOFF	5#													
HGHON	5#													
HIGHSW	5#													
HORPAG	5#													
IDX	5#	301	302	315										
ILLEGA	89	107	121	130	139	146	153	155	170	172	205#			
ILM	5#													
ILMENB	5#													
IMMB	5#	5												
INS	5#													
JDABLK	5#	227	267											
JOB41	39#	260												
JOBAPR	39#													
JOBDDT	39#													
JOBHRL	35#	325	327	334										
JOBPAG	5#	229	237											
JOBREN	39#													
JOBSA	39#													
JOBSYM	35#	323	336	337										
LOALOC	5#													
LOC	5#	307	309	311										
LSTCR	5													
MAXBLK	37#	297												
MEM41	246#	246												
MEMAPR	250#	250												
MEMBLK	36#	225	229	237	298									
MEMBLP	36#	226												
MEMDDT	248#	248												
MEMFF	254#	254												
MEMFIL	42#	131	147	157	175	222	223							
MEMGET	36#	228												
MEMINI	36#	224												
MEMORY	5#													
MEMREN	252#	252												
MEMSA	256#	256	281											
MEMSYM	36#	264												
MESDRP	20#	90	102	118	127	141	181	196						
METARS	38#	310												
MODUMP	42#	122	217	218	220									
MOT	29#	72	73	119	122	131	147							
MOTPTR	31#	73	92	93										
MSUITE	72#	123	132											
N1POIN	126	136#												
N2POIN	117	125#												
NADRB	5#	5												
NALPHA	86	98#												
NCOMMA	150	166#												
NMBPTR	32#	75	109	110	160	162								
NOCORE	171#	195												
NOFERM	167	179#												
NOMBRE	30#	74	75	113	114	157	163	175	208	209	210	211		
NONOUV	137	149#												
NONUME	100	116#												
NORPAG	5#													
NSUITE	74#	158	176											
NUMDRP	19#	88	108	112	154	171	183	198						
OPTESS	5	62	320											
ORGDRP	22#	120												
PAGBLN	35#	236	268	304										
PAGADR	37#	307	312											
PAGBLP	36#	277	278	290										
PAGEPC	39#	258	317											
PAGLNG	38#	234	305											
PAGLOC	5#	238	274	298	299	300	301							
PEEK	5#													
PIL	5#	58	63	156	161	174	224	228	283	284	285	286	291	294
	306	310	314	341										
PILPTR	35#	58												
PJGDRP	18#	87	104	138	151	168	185	194						
PREAB1	303	316#												
PREABS	302#	315												
PRGDRP	25#	152	169	187										

PRJDRP	21#	101	138	140	173									
RNGMEM	38#	314												
SD	5#													
SETODT	5#	263												
SETUWP	56#													
SLEEP	5#													
SMPTCH	63#	340												
SUITE	76#	94	111	115	142	148	184							
TAC	5#	5	55	56	60	61	68	69	216	218	219	220	221	223
	225	226	229	231	232	233	234	235	238	239	246	247	248	249
	250	251	252	253	254	255	256	257	258	259	260	261	263	264
	265	266	267	268	269	273	274	275	276	277	281	282	284	287
	288	289	290	298	299	323	324	326	327	328	330	331	334	335
	336	337	338	339	340									
TAC1	5#	227	235	236	237	289	292	293	295	296	297	325	326	328
	330	338												
TAC2	5#	308	313											
TAPMES	41#	346	349											
TTY	5#	64	70	71	215									
TTYHWI	42#	66	76	78										
US1AP0	41#	318												
US1DDT	40#	261												
USERGO	41#													
USR41	40#	247												
USRAPR	40#	251												
USRDA	40#	301												
USRDDT	40#	249	262											
USRFF	40#	255	282											
USRREL	37#	286												
USRREN	40#	253												
USRSA	40#	257	316											
USRSYM	39#	264	266	269										
USRUSY	39#	265												
UTILNG	38#	273	287											
UUO	5#	5												
UUOOFF	5#													
UUOON	5#													
UUOSW	5#	5												
ZTEMPR	35#	61												

X	241#	245

```
UCPROG - UNITE.CENTRALE.PROGRAMMEE POUR TRAVAIL EN PAGE 09.08.70          MACRO.V43 23:10  2-SEP-70 PAGE 1
CB     .MAC       DECLARATIONS GENERALES

                                ;CETTE FILE A ETE ASSEMBLEE AVEC LA FILE DES PARAMETRES COMMUNS
                                ;AUX SOUS-PROGRAMMES DE PAGINATION  CB.MAC

                                XLIST
                                IFDEF   LSTCB,<IFN LSTCB,<LIST>>

                                ;DEFINITION DES ACCUMULATEURS COMMUNS

                        000000  DRP==   0               ;DRAPEAUX (EN PARTIE GAUCHE)
                        000001  TAC==   1               ;TEMPORAIRE
                        000002  TAC1==  TAC+1           ;TEMPORAIRE FAIT SUITE A TAC POUR IDIV ET LSHC
                        000003  TAC2==  TAC+2           ;AUTRE TEMPORAIRE MOINS COURANT
                        000004  ADR==   4               ;CONTIENT L'ADRESSE ABSOLUE POUR ACCES MEMOIRE
                        000005  DAT==   5               ;TRANSMISSION D'ADRESSE
                        000006  CHN==   6               ;NUMERO DU CANAL OU DE L'ACCUMULATEUR POUR LES UUOS
                        000007  LOC==   7               ;ADRESSE EN MEMOIRE REELLE DE CE QUI FUT DEMANDE PAR (ADR)
                        000010  ACU==   10              ;NUMERO DE L'AC DE L'INSTRUCTION VIOLATRICE
                        000011  IDX==   11              ;QUANTITE D'INDEX POUR LES INSTRUCTIONS VIOLATRICES

                        000016  UUO==   16              ;CONTIENT L'UUO PENDANT SON ANALYSE
                        000016  INS==   UUO             ;CONTIENT L'INSTRUCTION EN CAS DE VIOLATION DE MEMOIRE
                        000017  PIL==   17              ;POINTEUR UNIQUE DE LA PILE DE TRAVAIL

                                ;NUMERO DES INSTRUCTIONS CALLI ET PARAMETRES ASSOCIES

                777777  777775  UUOSW==         -3      ;SUPPRESSION OU RETABLISSEMENT DES UUOS DU SYSTEME
                777777  777775  HIGHSW==        UUOSW   ;SUPRESSION OU RETABLISSEMENT DU SEGMENT HAUT
                        000000  UUOON==         0       ;RETABLISSEMENT DES UUOS
                        000001  UUOOFF==        1       ;SUPPRESSION DES UUOS
                        000002  HGHOFF==        2       ;SUPPRESSION DU SEGMENT HAUT
                        000003  HGHON==         3       ;RETABLISSEMENT DU SEGMENT HAUT
                        000002  SETDDT==        2       ;POUR PLACER L'ADRESSE DU DDT DANS LA ZONE PROTEGEE
                        000003  DDTOUT==        3       ;SORTIE D'UNE CHAINE ASCII+0 SUR LE TELETYPE
                        000001  DEVCHR==        1       ;OBTENTION DES CARACTERISTIQUES PHYSIQUE D'UN
                                                        ;PERIPHERIQUE
                        000011  CORE==          11      ;MODIFICATION DE L'ALLOCATION DE MEMOIRE
                        000012  EXIT==          12      ;SORTIE DU PROGRAMME-RETOUR AU MONITEUR
                        000016  ILMENB==        16      ;AUTORISATION DE PRISE EN COMPTE DES CAUSES D'ARRET
                                                        ;DU "CENTRAL-PROCESSOR" PAR EXEMPLE L'ILM,
                        020000  ILM==           20000   ;ILLEGAL-MEMORY-REFERENCE
                        000031  SLEEP==         31      ;SLEEP UNE SECONDE AU MOINS - POUR RESCHEDULING
                        000033  PEEK==          33      ;LIRE DANS LE MONITEUR

                                ;PARAMETRES RELATIFS AUX ENTREE-SORTIES

                        000000  TTY==           0       ;CANAL DU TTY AU MOMENT DE LA COMMANDE DE DEPART
                        000001  ASCIIL==        1       ;MODE ASCII+LIGNE POUR LE TELETYPE

                        000000  MEMORY==        0       ;NUMERO DE CANAL D'ACCES A LA MEMOIRE (DISQUE)
                        000017  DUMP==          17      ;MODE DUMP OU DIRECTEMENT AU MOYEN D'UNE LISTE
                                                        ;DE COMMANDE
                        000015  SD==            15      ;MODE DUMP POUR LE DISPLAY;NUMERO DE MODE DUMP
                                                        ;LE PLUS FAIBLE (UTILISE PAR UUOCON)
                        002000  PAGLOC==        2000    ;ADRESSE D'IMPLANTATION DU PROGRAMME UTILISATEUR
                        400000  LOALOC==        400000  ;ADRESSE ABSOLUE DE DEBUT DE CHARGEMENT
                        000001  JDABLK==        1       ;NOMBRE DE BLOCS DE JORDAT
                        000200  BLKSIZ==        200     ;NOMBRE DE MOTS PAR BLOC DE MEMOIRE ICI LE DISQUE
                777777  777771  BKSHFT==        -7      ;CONSTANTE DE DECALAGE BLKSIZ==2+BKSHFT

                                ;PAR CONVENTION AVEC LE CHARGEUR DE PAGINATION

                        000140  JOBPAG==        140     ;TAILLE ADOPTEE AU CHARGEMENT POUR LA BASE DES
                                                        ;PAGES (A GAUCHE) ET LA HAUTEUR DES PAGES (A DROITE)
                                ;DRAPEAUX DANS AC DRP
                        000001  IMMB==          1;MODE IMMEDIAT
                        000002  NADRB==         2       ;NE PAS DETRUIRE ADR
                        000003  ALAPRB==        NADRB!IMMB      ;TOUS
                        000004  HORPAG==        4       ;BESOIN DE SORTIR DE PAGE POUR L'INSTRUCTION
                        000010  NORPAG==        10      ;BESOIN DE MISE EN ABSOLU DE L'ADRESSE
                                IFNDEF  OPTESSAIS,<OPTESSAIS==-1>

                                LIST
                                TITLE   UCPROG - UNITE.CENTRALE.PROGRAMMEE POUR TRAVAIL EN PAGE 09.08.70
                                SUBTTL  DECLARATIONS GENERALES
                                ;ASSEMBLAGE AVEC LA FILE DES PARAMETRES CB.MAC

                                INTERN  US1AP1  ,US1AP0 ,VIOMM4 ,USRTRP

                                EXTERN  USRSAV  ,APRDRP ,USRXIT ,ILLINS ,ILEGAB ,ACCPAG ,DATREL ,RNGMEM
                                EXTERN  ACCUS   ,PAGEPC ,PINSDX ,PINSCD ,PINSAC ,PAGBLN ,PUROPE ,UUOSY1
                                EXTERN  PURARG  ,ACMMAB ,METABS ,ACCMM1 ,RNGMNB ,ACCMEM ,RNGMM1 ,PADRDX
                                EXTERN  ACCPAG  ,XCTOPE ,PADRDX ,PAGADR ,FORTY  ,JOBTPC ,JOBCNI ,USERGO
                                EXTERN  DATABS  ,USRAPR

                                ;DRAPEAUX PARTICULIERS A CETTE ROUTINE

                        000001  IMMB==  1               ;MODE IMMEDIAT
                        000002  NADRB== 2               ;NE PAS ECRASER ADR
                        000003  ALAPRB==                NADRB!IMMB      ;NETTOYAGE PREALABLE DES INDICATEURS
                        000004  HORPAG==4               ;INDIQUE BESOIN DE SORTIR DE PAGE

                        000002  MARGE== 2                ;CE QUE L'ON TOLERE COMME DEPASSEMENT
                                ;D'UNE PAGE DANS L'AUTRE (0 DANS 1) OU FAIT DE LA PROGRESSION NATURELLE
                                ;D'ADRESSE COURANTE

                777777  777777  LSTCB==-1               ;POUR PROVOQUER L'IMPRESSION DE CB.MAC
                                HLON
000000'                         HISEG
```

```
UCPROG - UNITE.CENTRALE.PROGRAMMEE POUR TRAVAIL EN PAGE 09.08.70          MACRO.V43 23:10  2-SEP-70 PAGE 2
UCPROG.MAC       CONTROLES PREALABLE A L'EXECUTION PROGRAMMEE

                                 SUBTTL  CONTROLES PREALABLE A L'EXECUTION PROGRAMMEE
                                 .ICI SUR INTERRUPTION ET TRAPPE EN POSITION 40

000000' 550040  000000* US1AP1: HRRZ    TAC,APRDRP       ;UTILISATEUR PEUT IL TRAITER CE GENRE D'ERREUR ?
000001' 620040  020000           TRZ     TAC,ILM          ;A L'EXCEPTION DE L'ILLEGAL MEMORY
000002' 672040  000000*          TDOE    TAC,JOBCNI       ; ?
000003' 254000  000105'          JRST    USRTRP           ;OUI...
                                 IFN     OPTESSAIS,<
000004' 606040  020000           TRNN    TAC,ILM          ;EST CE BIEN UNE ILM ?
000005' 254200  000006'          HALT    .+1>             ;NON ????
000006' 200200  000000*          MOVE    ADR,JOBTPC       ;COMPTEUR PROGRAMME QUAND INTERRUPTION
000007' 202200  000000*          MOVEM   ADR,PAGEPC       ;SAUVER EN PAGEPC POUR LA SUITE
000010' 201204  000000           MOVEI   ADR,(ADR)        ;PC SEUL
000011' 315200  000000*          CAMGE   ADR,PAG0LN       ; PC > PAGE 0 ?
000012' 254000  000053'          JRST    VIOMEM           ;NON, ERREUR EN ADRESSANT UNE DONNEE
000013' 550040  000011*          HRRZ    TAC,PAG0LN       ;MAXIMUM ADRESSABLE SANS INTERRUPTION
000014' 271040  000002           ADDI    TAC,MARGE        ;CE QUE L'ON TOLERE AU-DESSUS
000015' 303201  000000           CAILE   ADR,(TAC)        ;OUI-PAS UN PEU AU-DESSUS DE LA PAGE 0 ?
000016' 254000  000022'          JRST    US1AP0           ;NON - ADRESSE NETTEMENT ABSOLUE
000017' 275200  400000           SUBI    ADR,LOALOC       ;OUI - ADRESSE RELATIVE PAR +1 - OTER LE SEUIL
000020' 270200  000000*          ADD     ADR,PAGADR       ;PAGE SUIVANTE
000021' 542200  000007*          HRRM    ADR,PAGEPC       ;ADRESSE ABSOLUE POUR REPARTIR
000022' 260740  000000* US1AP0: PUSHJ   PIL,ACCPAG       ;ADRESSER LA PAGE NECESSITEE
000023' 201240  000021*          MOVEI   DAT,PAGEPC       ;POUR METTRE PAGEPC EN RELATIF
000024' 260740  000000*          PUSHJ   PIL,DATREL       ;...
000025' 254000  000000*          JRST    USERGO           ;ET RETOUR AVEC UNE PAGE NOUVELLE

                                 .VIOLATION DE MEMOIRE ALORS QUE PC EST DANS LA BONNE PAGE.CETTE PAGE
                                 .DEVRA ETRE LA MEME APRES LE TRAITEMENT DE L'INSTRUCTION

000026' 625200  777777  SETAL0: TLZA    ADR,777777       ;MASQUER LA PARTIE GAUCHE
000027' 550200  000023* SETALL: HRRZ    ADR,PAGEPC       ;PARTIE ADRESSE SEULE
000030' 260740  000000*          PUSHJ   PIL,ACMMAB       ;POUR AVOIR L'EMPLACEMENT DE L'INSTRUCTION
000031' 200707  000000           MOVE    INS,(LOC)        ;L'INSTRUCTION COMPLETE
000032' 135440  000000*          LDB     IDX,PINSDX       ;LE NO D'INDEX
000033' 332000  000011           SKIPE   IDX              ;INDEXE ?
000034' 540451  000000*          HRR     IDX,ACCUS(IDX)   ;OUI-PRENDRE L'INDEX
000035' 505440  000004           HRLI    IDX,ADR          ;POUR QUE @IDX==(IDX)+(ADR)
000036' 550200  000016           HRRZ    ADR,INS          ;ZONE ADRESSE RECOPIEE DANS ADR
000037' 135400  000000*          LDB     ACU,PINSAC       ;NO DE L'ACCUMULATEUR
000040' 603700  000020           TLNE    INS,20           ;ADRESSAGE INDIRECT ?
000041' 661440  400000           TLO     IDX,1818         ;BIT DE SIGNE DE IDX SI INDIRECT
000042' 135040  000000*          LDB     TAC,PINSCD       ;CODE D'INSTRUCTION
000043' 646040  000700           TRCN    TAC,700          ;UUO ?
                                 JRST    [MOVEI  TAC,-1           ;OUI - A PART
000044' 254000  000557'                  POPJ    PIL,0]           ;RETOUR
000045' 646040  000700           TRCN    TAC,700          ;I/O INSTRUCTION ?
                                 JRST    [MOVEI  TAC,-2           ;OUI - ILLEGAL EN MODE UTILISATEUR
000046' 254000  000561'                  POPJ    PIL,0]           ;RETOUR
000047' 275040  000100           SUBI    TAC,100          ;POUR LES UUOS SUPPRIMEES
000050' 231040  000007           IDIVI   TAC,<+D36/UU>    ;POUR DISPATCH SELON LE CODE INSTRUCTION
000051' 135042  000461'          LDB     TAC,INSPTR(TAC1);AU MOYEN DE LA TABLE
000052' 263740  000000* SETRET: POPJ    PIL,USRXIT       ;RETOUR AVEC CODE DE DISPATCH

000053' 201240  000027* VIOMEM: MOVEI   DAT,PAGEPC       ;POUR METTRE DAT EN ABSOLU
000054' 260740  000000*          PUSHJ   PIL,DATARS       ;...
000055' 621000  000003           TLZ     DRP,ALAPRB       ;PRECAUTION
000056' 370200  000053*          SOS     ADR,PAGEPC       ;VERIFIER ADRESSE -1
000057' 260740  000027'          PUSHJ   PIL,SETALL       ;PREPARER LES REGISTRES ET CODE DE DISPATCH
000060' 201100  400000           MOVEI   TAC1,LOALOC      ;PAS D'AMBIGUITE SI EN TETE DE PAGE
000061' 430100  000056*          XOR     TAC1,PAGEPC      ;COMPARER AU COMPTEUR PROGRAMME
000062' 602100  777777           TRNE    TAC1,-1          ;IDENTITE ?
000063' 302040  000000           CAIE    TAC,WRT          ;INSTRUCTION DU GENRE ECRITURE ?
000064' 325440  000076'          JUMPGE  IDX,VIOMM3       ;NON ETUDIER L'INSTRUCTION POINTEE PAR JOBTPC
000065' 621000  000004           TLZ     DRP,HORPAG       ;POUR VERIFIER SI NECESSITE D'ALLER HORS PAGE
000066' 661000  000001           TLO     DRP,IMMB         ;SIMPLEMENT CALCULER 0-E
000067' 260740  000405'          PUSHJ   PIL,ACCESM       ;...CALCUL DE L'ADRESSE
000070' 603000  000004  VIOMM1: TLNE    DRP,HORPAG       ;BESOIN D'ALLER HORS PAGE ?
000071' 254000  000077'          JRST    VIOMM4           ;OUI , LA TRAPPE AURAIT EU LIEU EN . ET PAS .+1
000072' 661000  000002           TLO     DRP,NADRA        ;ACCEDER A LA QUANTITE
000073' 621000  000001           TLZ     DRP,IMMB         ;DEJA FAIT ?
000074' 260740  000417'          PUSHJ   PIL,ACCES1       ;NON , ACCEDER A LA QUANTITE
000075' 607000  000004           TLNN    DRP,HORPAG       ;VIOLATION POUR L'ACCES A LA DONNEE ?
000076' 350200  000061* VIOMM3: AOS     ADR,PAGEPC       ;NON - REMETTRE A LA VALEUR COURANTE
000077' 261740  000052' VIOMM4: PUSH    PIL,SETRET       ;POUSSER L'ADRESSE DE RETOUR SUR LA PILE
000100' 260740  000027'          PUSHJ   PIL,SETALL       ;PREPARER LES REGISTRES ET CODE DE DISPATCH
000101' 621000  000003  VIOMM5: TLZ     DRP,ALAPRB       ;NETTOYAGE PREALABLE DES INDICATEURS POSSIBLES
000102' 325441  000112'          JUMPGE  IDX,DISPAT(TAC)  ;DISPATCH ET SORTIE PAR POPJ PIL,0
000103' 260741  000112'          PUSHJ   PIL,DISPAT(TAC)  ;VERIFIER ENSUITE L'INSTRUCTION SUIVANTE
000104' 254000  000076'          JRST    VIOMM3           ;JUSQU'A PLUS D'INDIRECTION

                                 .RENVOI A L'UTILISATEUR DES INTERRUPTIONS VRAIES OU FAUSSES DONT
                                 .IL A DEMANDE LE CONTROLE PAR CALLI AC,16

000105' 540040  000000* USRTRP: HRR     TAC,USRAPR       ;SON ADRESSE DE RENVOI
000106' 542040  000076*          HRRM    TAC,PAGEPC       ;POUR RETOUR A L'UTILISATEUR
000107' 254000  000022'          JRST    US1AP0           ;...
```

```
                                SUBTTL  ROUTINES SPECIALISEES SELON TYPE D'INSTUCTION
                                DEFINE  GENRE,<
                                        X       WRT
                                        X       PSJ
                                        X       PSH
                                        X       POP
                                        X       JSR
                                        X       JSP
                                        X       JRS
                                        X       RED
                                        X       IMM
                                        X       RMW
                                        X       BYT
                                        X       NXI
                                        X       BRK
                                        X       PTR
                                        X       NOP
                                        X       PPJ
                                >
                                DEFINE  X (A),<
                                        JRST    A'DSP
                                        A**     .-1-DISPAT
                                >
000110' 254000  000000*                 JRST    ILLINS          ;-2 ILLEGAL >700
000111' 254000  000377'                 JRST    VIOUUO          ;-1 UUO     <100
000112'                         DISPAT: GENRE*
000112' 254000  000134'                 JRST    WRTDSP
000113' 254000  000260'                 JRST    PSJDSP
000114' 254000  000277'                 JRST    PSHDSP
000115' 254000  000322'                 JRST    POPDSP
000116' 254000  000245'                 JRST    JSRDSP
000117' 254000  000252'                 JRST    JSPDSP
000120' 254000  000254'                 JRST    JRSDSP
000121' 254000  000132'                 JRST    REDDSP
000122' 254000  000147'                 JRST    IMMDSP
000123' 254000  000141'                 JRST    RMWDSP
000124' 254000  000153'                 JRST    BYTDSP
000125' 254000  000287'                 JRST    NXIDSP
000126' 254000  000216'                 JRST    BRKDSP
000127' 254000  000355'                 JRST    PTRDSP
000130' 254000  000243'                 JRST    NOPDSP
000131' 254000  000337'                 JRST    PPJDSP
                        000005  UU**   *D36-*L<.-DISPAT>


                                ;TRAITEMENT DES INSTRUCTIONS DE LECTURE SEULEMENT

000132' 260740  000405' REDDSP: PUSHJ   PIL,ACCESM      ;ACCEDER AU TERME INDIQUER PAR L'ADRESSE
000133' 254000  000443'         JRST    XCTINS          ;EXECUTER L'INSTRUCTION ET EXIT

                                ;TRAITEMENT DES INSTRUCTIONS D'ECRITURE OU DE LECTURE PUIS REECRITURE

000134' 335000  000011  WRTDSP: SKIPGE  IDX             ;INDIRECTION ?
000135' 262740  000001          POP     PIL,TAC         ;OUI - CORRIGER LA PILE POUR RETOUR
000136' 260740  000141'         PUSHJ   PIL,RMWDSP      ;EXECUTER L'INSTRUCTION
000137' 262740  000001          POP     PIL,TAC         ;RECTIFIER LA PILE (ADRESSE DE RETOUR)
000140' 254000  000076'         JRST    VIOMM3          ;EXECUTER L'INSTRUCTION SUIVANTE

000141' 661000  000002  RMWDSP: TLO     DRP,NADRR       ;GARDER L'ADRESSE
000142' 260740  000405'         PUSHJ   PIL,ACCESM      ;ACCEDER AU TERME INDIQUE PAR L'ADRESSE
000143' 261740  000004          PUSH    PIL,ADR         ;SAUVER POUR LE RNGMEM
000144' 260740  000443'         PUSHJ   PIL,XCTINS      ;EXECUTER L'INSTRUCTION
000145' 262740  000004          POP     PIL,ADR         ;RESTAURER
000146' 254000  000000*         JRST    RNGMEM          ;RANGER AU BESOIN LE BLOC OU L'ON A ECRIT ET EXIT

                                ;TRAITEMENT DES INSTRUCTIONS A QUANTITE IMMEDIATE

000147' 661000  000003  IMMDSP: TLO     DRP,IMMB!NADRB  ;LECTURE IMMEDIATE
000150' 260740  000405'         PUSHJ   PIL,ACCESM      ;ACCEDER A LA QUANTITE IMMEDIATE (CALCULER)
000151' 541704  000000          HRRI    INS,(ADR)       ;0-E=(ADR)+(IDX) EVENTUEL
000152' 254000  000444'         JRST    XCTIN1          ;ET EXECUTER PUIS EXIT

                                ;MANIPULATION DE BYTES

000153' 205040  020000  BYTDSP: MOVSI   TAC,(1B4)       ;BYTE INTERRUPT FLAG
000154' 612040  000106*         TDNE    TAC,PAGEPC      ;EST-IL POSITIONNE ?
000155' 661700  001000          TLO     INS,(1B8)       ;OUI - NE PAS INCREMENTER LE POINTEUR
000156' 412040  000154*         ANDCAM  TAC,PAGEPC      ;ET NETTOYER LE BIT EN QUESTION
000157' 661000  000002          TLO     DRP,NADRR       ;POUR CONSERVER ADR ABSOLU
000160' 260740  000405'         PUSHJ   PIL,ACCESM      ;ACCES AU POINTEUR
000161' 261740  000004          PUSH    PIL,ADR         ;SAUVER ADRESSE DU POINTEUR
000162' 200207  000000          MOVE    ADR,(LOC)       ;LE POINTEUR INDIQUE L'ADRESSE
000163' 603700  004000          TLNE    INS,(1B6)       ;EST-CE IBP ?
000164' 667700  001000          TLON    INS,(1B8)       ;NON-EST CE DPB OU LDB ?
000165' 133000  000004          IBP     ADR             ;OUI-NON C'EST ILDB,IBP OU IDPB
000166' 202207  000000          MOVEM   ADR,(LOC)       ;RANGER A LA MEME PLACE LE POINTEUR INCREMENTE
000167' 250217  000000          EXCH    ADR,(PIL)       ;SAUVER LE POINTEUR EVENTUELLEMENT CHANGE
000170' 260740  000146*         PUSHJ   PIL,RNGMEM      ;RANGER LE POINTEUR S'IL LE FAUT
000171' 200217  000000          MOVE    ADR,(PIL)       ;POINTEUR
000172' 260740  000425'         PUSHJ   PIL,ACCES2      ;ACCEDER A L'ARGUMENT COMME PAR UNE INDIRECTION
000173' 250217  000000          EXCH    ADR,(PIL)       ;RECUPERER LE POINTEUR
000174' 621200  000037          TLE     ADR,37          ;PLUS D'INDEX NI D'INDIRECTION
000175' 541207  000000          HRRI    ADR,(LOC)       ;ADRESSE LOCALE
000176' 201040  000001          MOVEI   TAC,TAC         ;TRAVAILLER AVEC L'AC STANDARD
000177' 137040  000037*         DPB     TAC,PINSAC      ;DANS INS...
000200' 541700  000004          HRRI    INS,ADR         ;ADRESSE DU POINTEUR
000201' 200050  000034*         MOVE    TAC,ACCUS(ACU)  ;L'AC
000202' 621700  000037          TLE     INS,37          ;PLUS D'INDIRECTION NI D'INDEXATION
000203' 256000  000016          XCT     INS             ;FAIRE LA MANIPULATION DE BYTE
000204' 202050  000201*         MOVEM   TAC,ACCUS(ACU)  ;RANGER L'AC
000205' 262740  000004          POP     PIL,ADR         ;ADRESSE DU MOT POINTE
000206' 254000  000170*         JRST    RNGMEM          ;RANGER LE BLOC EN MEMOIRE ET EXIT
                                ;INSTRUCTIONS QUI N'EXISTENT PAS

000207' 201000  000211' NXIDSP: MOVEI   NXIMES          ;ADRESSE DU MESSAGE
000210' 254000  000000*         JRST    ILEGAB          ;IMPRIMER AVEC L'ADRESSE ABSOLUE

000211' 446352  352244  NXIMES: ASCIZ   "INSTRUCTION INEXISTANTE"
000212' 526072  444636
000213' 471011  147212
000214' 542232  352202
000215' 472510  500000
```

```
                                ;INSTRUCTIONS DE RUPTURE (EVENTUELLE) DE SEQUENCE

000216' 260740  000455' BRKDSP: PUSHJ   PIL,SETPAG       ;CHARGER LA PAGE NECESSAIRE AU SAUT
000217' 541704  000000          HRRI    INS,(ADR)        ;***** ?
000220' 621700  000037          TLZ     INS,37           ;PLUS D'INDIRECTION NI D'INDEXATION
000221' 201040  000204*         MOVEI   TAC,ACCUS        ;ATTENTION AUX AC'S ADRESSES
000222' 307056  000000          CAIG    TAC,0(INS)
000223' 307056  777760          CAIG    TAC,-20(INS)
000224' 334000  000000          SKIPA                    ;PAS AC
000225' 275701  000000          SUBI    INS,(TAC)        ;SUPPRIMER LE SEUIL DES AC'S
000226' 202700  000000*         MOVEM   INS,PUROPE
000227' 201240  000226*         MOVEI   DAT,PUROPE       ;LA PAGE EST BONNE - METTRE EN RELATIF
000230' 260740  000024*         PUSHJ   PIL,DATRFL       ;...
000231' 641700  256000          TLC     INS,(XCT)        ;RECHERCHE DE XCT
000232' 607700  777000          TLNN    INS,777000       ;EST - CE ?
                                JRST    [PUSHJ PIL,SETALO        ;ADR EST PRET
000233' 254000  000563'                 JRST    VIOHM5]          ;NOUVELLE INSTRUCTION
000234' 550040  000156*         HRRZ    TAC,PAGEPC       ;ADRESSE DE L'INSTRUCTION
000235' 271040  000001          ADDI    TAC,1            ; .+1
000236' 505040  254000          HRLI    TAC,(JRST)
000237' 202040  000001*         MOVEM   TAC,PUROPE+1     ;EN CAS DE NON-RUPTURE
000240' 541040  000227*         HRRI    TAC,PUROPE       ;
000241' 542040  000234*         HRRM    TAC,PAGEPC       ;L'EXIT SE FERA LA
000242' 254000  000025*         JRST    USERGO           ;DIRECTEMENT

                                ;INSTRUCTIONS NE NECESSITANT PAS D'ACCES MEMOIRE POUR LEUR EXECUTION

000243' 201340  000000  NOPDSP: MOVEI   LOC,0            ;ADRESSE =0
000244' 254000  000443'         JRST    XCTINS           ;L'EXECUTER QUAND MEME ET EXIT

                                ;INSTRUCTIONS PROGRAMMEES JSR <CODE==105> ET JSP <CODE==106>

000245' 260740  000455' JSRDSP: PUSHJ   PIL,SETPAG       ;BONNE PAGE POUR LE SAUT
000246' 350040  000241*         AOS     TAC,PAGEPC       ;PC + 1 POUR LE RETOUR
000247' 202047  000000          MOVEM   TAC,(LOC)        ;RANGER
000250' 542200  000246*         HRRM    ADR,PAGEPC       ;SERA INCREMENTE EN FIN DE TRAITEMENT DE L'INS
000251' 263740  000000          POPJ    PIL,0            ;FIN DE L'INSTRUCTION JSR
000252' 350040  000250* JSPDSP: AOS     TAC,PAGEPC       ;ADRESSE DE RETOUR ET FLAGS
000253' 202050  000221*         MOVEM   TAC,ACCUS(ACU)   ;RANGEMENT DANS LE BON ACCUMULATEUR
000254' 260740  000455' JRSDSP: PUSHJ   PIL,SETPAG       ;BONNE PAGE POUR LE SAUT
000255' 275200  000001          SUBI    ADR,1            ;A CAUSE DE L'INCREMENTATION SYSTEMATIQUE
000256' 542200  000252*         HRRM    ADR,PAGEPC       ;RANGEMENT DE L'ADRESSE DE BRANCHEMENT
000257' 263740  000000          POPJ    PIL,0            ;FIN DE L'INSTRUCTION JSP

                                ;INSTRUCTION PUSHJ

000260' 350050  000253* PSJDSP: AOS     TAC,ACCUS(ACU)   ;POINTEUR
000261' 577000  000001          HLRES   TAC              ;COMPTE DE MOTS
000262' 342040  000310'         AOJE    TAC,PDLOVF       ;VERIFIER LE PDL OVER-FLOW
000263' 506050  000260*         HRLM    TAC,ACCUS(ACU)   ;TRES BIEN REMETTRE DANS L'AC
000264' 550210  000263*         HRRZ    ADR,ACCUS(ACU)   ;POINTEUR VERS LA PILE
000265' 260740  000030*         PUSHJ   PIL,ACMMAB       ;ACCES APRES MISE EN ABSOLU DE L'ADRESSE
000266' 200040  000256*         MOVE    TAC,PAGEPC       ;FLAGS ET COMPTEUR PROGRAMME EN ABSOLU
000267' 271040  000001          ADDI    TAC,1            ;PROGRESSION NORMALE
000270' 202047  000000          MOVEM   TAC,(LOC)        ;SUR LA PILE
000271' 260740  000206*         PUSHJ   PIL,RNGMEM       ;SAUVER L'ETAT DE LA PILE
000272' 550200  000016          HRRZ    ADR,INS          ;CALCUL DE L'ADRESSE OU SAUTER
000273' 260740  000455'         PUSHJ   PIL,SETPAG       ;ALLER CHERCHER L'ADRESSE
000274' 275200  000001          SUBI    ADR,1            ;A CAUSE DE L'INCREMENTATION SYSTEMATIQUE
000275' 542200  000266*         HRRM    ADR,PAGEPC       ;DE PAGEPC
000276' 263740  000000          POPJ    PIL,0            ;EXIT RETOUR A L'UTILISATEUR

                                ;INSTRUCTION PUSH

000277' 260740  000405' PSHDSP: PUSHJ   PIL,ACCESM        ;ACCEDER A LA QUANTITE
000300' 200447  000000          MOVE    IDX,(LOC)        ;CONSERVER LA ;LA QUANTITE
000301' 550210  000264*         HRRZ    ADR,ACCUS(ACU)   ;ADRESSE POINTEE
000302' 260740  000000*         PUSHJ   PIL,METABS       ;MISE EN ABSOLU
000303' 260740  000000*         PUSHJ   PIL,ACCMM1       ;ACCEDER A L'ADRESSE+1
000304' 202447  000000          MOVEM   IDX,(LOC)        ;Y RANGER LA QUANTITE
000305' 350050  000301*         AOS     TAC,ACCUS(ACU)   ;INCREMENTER LE POINTEUR DE PILE
000306' 577000  000001          HLRES   TAC              ;LE COMPTE DE MOTS
000307' 346040  000335'         AOJN    TAC,PPHDS1       ;INCREMENTER ET L'ALLER RANGER SI NON NUL
000310' 201100  202000  PDLOVF: MOVEI   TAC1,1B19!1B25    ;PDLOVF ET CLOCK ENABLE
000311' 550040  000000*         HRRZ    TAC,APRDRP       ;DRAPEAUX DE L'APR DESIRES PAR L'UTILISATEUR
000312' 202100  000002*         MOVEM   TAC1,JOBCNI      ;COMME UNE INTERRUPTION DE L'APR
000313' 602040  200000          TRNE    TAC,1B19         ;AUTORISE A TRAITER LUI MEME SES PDLOVF ?
000314' 254000  000105'         JRST    USRTRP           ;OUI...
000315' 201000  000317'         MOVEI   PILMES           ;ADRESSE DE LA CHAINE ASCIZ
000316' 254000  000210*         JRST    ILEGAB           ;IMPRIMER AVEC L'ADRESSE ABSOLUE
000317' 502111  420236  PILMES: ASCIZ   .PDL OVERFLOW.
        000320' 532132  243230
        000321' 476560  000000
                                ;INSTRUCTION POP

000322' 550210  000305* POPDSP: HRRZ    ADR,ACCUS(ACU)   ;ADRESSE DE LA PILE
000323' 260740  000265*         PUSHJ   PIL,ACMMAB       ;ACCES APRES MISE EN ABSOLU DE L'ADRESSE
000324' 261747  000000          PUSH    PIL,(LOC)        ;PRENDRE LA QUANTITE SUR LA PILE
000325' 661000  000002          TLO     DRP,NADRB        ;POUR CONSERVER ADR
000326' 201216  000000          MOVEI   ADR,(INS)        ;ADRESSE DONNEE PAR L'INSTRUCTION
000327' 260740  000405'         PUSHJ   PIL,ACCESM       ;ACCEDER A CETTE POSITION
000330' 262747  000000          POP     PIL,(LOC)        ;Y RANGER LA QUANTITE
000331' 370050  000322*         SOS     TAC,ACCUS(ACU)   ;DECREMENTER LA PILE
000332' 577000  000001          HLRES   TAC              ;COMPTE DE MOTS
000333' 322040  000310'         JUMPE   TAC,PDLOVF       ;SI 0 AVANT DECREMENTATION -1 APRES
000334' 370000  000001          SOS     TAC              ;DECREMENTER
000335' 506050  000331* PPHDS1: HRLM    TAC,ACCUS(ACU)   ;RANGER DANS L'AC DE L'UTILISATEUR
000336' 254000  000271*         JRST    RNGMEM           ;RANGER EN MEMOIRE ET EXIT

                                ;INSTRUCTION POPJ

000337' 550210  000335* PPJDSP: HRRZ    ADR,ACCUS(ACU)   ;ADRESSE OU TROUVER L'ADRESSE DE RETOUR
000340' 260740  000323*         PUSHJ   PIL,ACMMAB       ;ACCES A LA PILE APRES MISE EN ABSOLU
000341' 200047  000000          MOVE    TAC,(LOC)        ;ADRESSE DE RETOUR
000342' 621040  000037          TLZ     TAC,37           ;PAS D'INDEX NI D'INDIRECTION (POPJ)
000343' 202040  000275*         MOVEM   TAC,PAGEPC       ;RANGER POUR SAUTER LA , EN SORTIE
000344' 201201  777777          MOVEI   ADR,-1(TAC)      ;RECOPIE DE L'ADRESSE -1 (SERA INCREMENTE)
000345' 260740  000302*         PUSHJ   PIL,METABS       ;ASSURER L'ABSOLU
000346' 542200  000343*         HRRM    ADR,PAGEPC       ;RANGER AVANT SORTIE
000347' 370050  000337*         SOS     TAC,ACCUS(ACU)   ;DECREMENTER LE POINTEUR DE PILE
000350' 577000  000001          HLRES   TAC              ;LE COMPTE
000351' 322040  000310'         JUMPE   TAC,PDLOVF       ;SI 0 MAINTENANT , -1 APRES DECREMENTATION
000352' 370000  000001          SOS     TAC              ;DECREMENTER LE COMPTE
000353' 506050  000347*         HRLM    TAC,ACCUS(ACU)   ;REMETTRE DANS L'AC UTILISATEUR
000354' 263740  000000          POPJ    PIL,0            ;ET EXIT
```

```
                                ;INSTRUCTION DE TRANSFERT DE BLOCS 'BLT'

000355' 661000  000002  PTRDSP: TLO     DRP,NADRB       ;CALCUL D'ADRESSE SEULEMENT
000356' 260740  000485'         PUSHJ   PIL,ACCESM      ;ACCEDER A LA LIMITE
000357' 201444  000000          MOVEI   IDX,(ADR)       ;SAUVER L'ADRESSE ABSOLUE
000360' 554210  000353*         HLRZ    ADR,ACCUS(ACU)  ;ADRESSE SOURCE
000361' 550410  000360*         HRRZ    ACU,ACCUS(ACU)  ;ADRESSE DESTINATION
000362' 260740  000340*         PUSHJ   PIL,ACMMAB      ;ACCES A LA SOURCE APRES MISE EN ABSOLU
000363' 261747  000000          PUSH    PIL,(LOC)       ;SAUVIR LA QUANTITE A MOUVOIR
000364' 250200  000010          EXCH    ADR,ACU         ;IDEM POUR LA DESTINATION
000365' 260740  000362*         PUSHJ   PIL,ACMMAB      ;ACCEDER A LA DESTINATION APRES MISE EN ABSOLU
000366' 262747  000000  BLTIN0: POP     PIL,(LOC)       ;DEPOSER EN BONNE PLACE
000367' 306211  000000          CAIN    ADR,(IDX)       ;COMPARER A L'ADRESSE ABSOLUE LIMITE DU BLT
000370' 254000  000336*         JRST    RNGMEM          ;TERMINE , RANGER LE DERNIER BLOC UTILISE
000371' 250200  000010          EXCH    ADR,ACU         ;REPASSER A L'ADRESSE SOURCE
000372' 260740  000000*         PUSHJ   PIL,RNGMM1      ;ACCES A L'ADRESSE SUIVANTE EN RANGEANT AU BESOIN
000373' 261747  000000          PUSH    PIL,(LOC)       ;SAUVER LA QUANTITE A MOUVOIR
000374' 250200  000010          EXCH    ADR,ACU         ;REPASSER A L'ADRESSE DESTINATION
000375' 260740  000372*         PUSHJ   PIL,RNGMM1      ;ACCES A L'ADRESSE SUIVANTE POUR RANGER
000376' 254000  000366'         JRST    BLTIN0          ;ET BOUCLER

                                ;CAS DES UUOS (LE CALCUL DE L'ADRESSE SEUL A PU PROVOQUER LA VIOLATION DE MEMOIRE)

000377' 661000  000001  VIOUUO: TLO     DRP,IMMB        ;PAS D'ACCES A LA QUANTITE ADRESSEE
000400' 260740  000485'         PUSHJ   PIL,ACCESM      ;CALCUL D'ADRESSE
000401' 541704  000000          HRRI    INS,(ADR)       ;QUANTITE ABSOLUE
000402' 621700  000037          TLZ     INS,37          ;PLUS D'INDEXATION NI D'INDIRECTION
000403' 202700  000000*         MOVEM   INS,FORTY       ;COMME UNE UUO TRAPPEE
000404' 254000  000000*         JRST    UUOSY1          ;REJOINDRE DE SUITE LA ROUTINE DES UUOS


                                SUBTTL  ROUTINES COMMUNES
                                        ;ROUTINES COMMUNES AUX DIFFERENTS TYPES D'INSTRUCTIONS
                                ;ROUTINE D'ACCES A UNE QUANTITE OU A UNE ADRESSE (QUANTITE IMMEDIATE)

000405' 260740  000345* ACCESM: PUSHJ   PIL,METABS      ;MISE EN ABSOLU DE ADR SI APPAREMMENT TRANSLATABLE
000406' 506217  000000          HRLM    ADR,(PIL)       ;CONSERVATION
000407' 201211  000000          MOVEI   ADR,(IDX)       ;RECOPIE DE L'INDEX POUR MISE EN ABSOLU
000410' 260740  000405*         PUSHJ   PIL,METABS      ;...
000411' 544457  000000          HLR     IDX,(PIL)       ;RECUPERER L'ANCIEN ADR
000412' 201220  000011          MOVEI   ADR,@IDX        ;(ADR)←(ADR)+(IDX) TIENT COMPTE DE L'INDEXATION
000413' 603000  000001  ACCES0: TLNE    DRP,IMMB        ;MODE IMMEDIAT ?
000414' 603440  400000          TLNE    IDX,1B18        ;OUI - ET NON INDIRECT ?
000415' 254000  000417'         JRST    ACCES1          ;NON
000416' 263740  000000          POPJ    PIL,0           ;EXIT POUR L'IMMEDIAT SANS INDIRECTION
000417' 260740  000000* ACCES1: PUSHJ   PIL,ACCMEM      ;NON - IL FAUT ACCEDER AU MOT POINTE PAR ADR
000420' 607000  000002          TLNN    DRP,NADRB       ;PEUT-ON DETRUIRE L'ADRESSE ?
000421' 200207  000000          MOVE    ADR,(LOC)       ;OUI
000422' 627440  400000          TLZN    IDX,1B18        ;ETAIT-ON EN MODE INDIRECT ?
000423' 263740  000000          POPJ    PIL,0           ;NON - ADR CONTIENT SANS DOUTE LA QUANTITE
000424' 200207  000000          MOVE    ADR,(LOC)       ;EN CAS DE NADRB
000425' 135440  000000* ACCES2: LDB     IDX,PADRDX      ;ZONE INDEX DU NOUVEAU MOT
000426' 332000  000011          SKIPE   IDX             ;PAS SI 0
000427' 540451  000361*         HRR     IDX,ACCUS(IDX)  ;VALEUR D'INDEX
000430' 661440  000004          TLO     IDX,ADR         ;POUR INDEXER PAR ADR
000431' 603200  000020          TLNE    ADR,20          ;NOUVELLE ADRESSE INDIRECTE ?
000432' 661440  400000          TLO     IDX,1B18        ;OUI - MARQUER PAR LE BIT DE SIGNE
000433' 201220  000011          MOVEI   ADR,@IDX        ;INDEXATION EVENTUELLE
000434' 261740  000020*         PUSH    PIL,PAGADR      ;METTRE EN ABSOLU COMPTE TENU DE LA SOURCE
000435' 557000  000007          HLRZS   LOC             ;...
000436' 322340  000440'         JUMPE   LOC,.+2         ;CAS DES REGISTRES COMME SOURCE
000437' 202340  000434*         MOVEM   LOC,PAGADR      ;POUR UTILISER METABS
000440' 260740  000410*         PUSHJ   PIL,METABS      ;MISE EN ABSOLU DE ADR
000441' 262740  000437*         POP     PIL,PAGADR      ;RETABLIR
000442' 254000  000413'         JRST    ACCES0          ;BOUCLE APRES INDIRECTION

                                ;EXECUTION DE L'INSTRUCTION

000443' 541707  000000  XCTINS: HRRI    INS,(LOC)       ;ADRESSE LOCALE
000444' 621700  000037  XCTIN1: TLZ     INS,37          ;INDIRECTION ET INDEXATION DEJA PRISES EN COMPTE
000445' 202700  000240*         MOVEM   INS,PUROPE      ;EN MEMOIRE POUR XCT
000446' 621700  777777          TLZ     INS,-1          ;ATTENTION AUX ADRESSES D'ACS
000447' 301700  000427*         CAIL    INS,ACCUS
000450' 301700  000020*         CAIL    INS,ACCUS+20
000451' 254000  000000*         JRST    XCTOPE          ;NON - ALLER COMME CELA
000452' 275700  000447*         SUBI    INS,ACCUS       ;OTER LE SEUIL
000453' 542700  000445*         HRRM    INS,PUROPE      ;AVANT EXECUTION
000454' 254000  000451*         JRST    XCTOPE          ;RESTAURE LES AC'S DE L'UTILISATEUR
                                                        ;NE SAUVE QUE PIL ET EXIT PAR POPJ


                                ;POUR METTRE LA PAGE INDIQUEE  AR L'ADRESSE DE L'INSTRUCTION

000455' 661000  000001  SETPAG: TLO     DRP,IMMB        ;IMMEDIAT
000456' 260740  000485'         PUSHJ   PIL,ACCESM      ;CALCUL DE L'ADRESSE
000457' 260740  000440*         PUSHJ   PIL,METABS      ;ASSURER QU'ELLE EST EN ABSOLU
000460' 254000  000022*         JRST    ACCPAG          ;ET ENTRER LA PAGE CONCERNEE
```

```
                                SUBTTL  TABLEAU DE DEFINITION DES GENRES D'INSTRUCTIONS

                        000001  ZZ==1
                                INSPTR: REPEAT  <+D36/UU>,<POINT UU,INSTAB(TAC),<<UU*ZZ>-1>
                                                ZZ=ZZ+1>
000461' 370501  000470' POINT UU,INSTAB(TAC),<<UU*ZZ>-1>
                        000002                  ZZ=ZZ+1
000462' 320501  000470' POINT UU,INSTAB(TAC),<<UU*ZZ>-1>
                        000003                  ZZ=ZZ+1
000463' 250501  000470' POINT UU,INSTAB(TAC),<<UU*ZZ>-1>
                        000004                  ZZ=ZZ+1
000464' 200501  000470' POINT UU,INSTAB(TAC),<<UU*ZZ>-1>
                        000005                  ZZ=ZZ+1
000465' 130501  000470' POINT UU,INSTAB(TAC),<<UU*ZZ>-1>
                        000006                  ZZ=ZZ+1
000466' 060501  000470' POINT UU,INSTAB(TAC),<<UU*ZZ>-1>
                        000007                  ZZ=ZZ+1
000467' 010501  000470' POINT UU,INSTAB(TAC),<<UU*ZZ>-1>
                        000010                  ZZ=ZZ+1

                                DEFINE  Y (A)   ,<IRP A
                                        <IFE    ZZ-1,<ZZZ==0>
                                        ZZZ==ZZZ!<A-<+D36-<ZZ*UU>>>
                                        ZZ==ZZ+1
                                        IFG     ZZ*UU-+D36,<ZZ==1
                                        EXP     ZZZ
                                        >>>
                        000001  ZZ==1

                                ;LA TABLE INSTAB INDIQUE LE GENRE D'UTILISATION DE LA MEMOIRE
                                ;QUE FAIT CHAQUE INSTRUCTION MACHINE (CODES 100 A 677(8)).

                                ;ATTENTION- EN CAS DE CORRECTION DES ARGUMENTS DE "Y" ;LES LIGNES UTILES
                                ;COMMENCANT PAR UNE VIRGULE SE TERMINENT TOUTES PAR UN INDICATEUR DE CONTINUATION
                                ;DE LIGNE (+- OU CONTROL-SHIFT-"O")AU MOINS 1 FOIS + AUTANT QUE DE LIGNES
                                ;DE COMMENTAIRES (;....).

                                INSTAB: Y       <
                                ;CODES 100 A 107 N'EXISTENT PAS
                                 NXI    ,PSJ    ,PPJ    ,PSH    ,POP    ,JSR    ,JSP    ,NXI
                                ;
                                ;CODES 110 A 117 N'EXISTENT PAS
                                ,NXI    ,NXI    ,NXI    ,NXI    ,NXI    ,NXI    ,NXI    ,NXI
                                ;
                                ;CODES 120 A 127 N'EXISTENT PAS
                                ,NXI    ,NXI    ,NXI    ,NXI    ,NXI    ,NXI    ,NXI    ,NXI
                                ;
                                ;UFA    ,DFN    ,FSC    ,IBP    ,ILDB   ,LDB    ,IDPB   ,DPB
                                ,RED    ,RMW    ,IMM    ,BYT    ,BYT    ,BYT    ,BYT    ,BYT
                                ;
                                ;FAD    ,FADL   ,FADM   ,FADR   ,FADR   ,FADRI  ,FADRM  ,FADRB
                                ,RED    ,RED    ,RMW    ,RMW    ,RED    ,IMM    ,RMW    ,RMW
                                ;
                                ;FSB    ,FSBL   ,FSBM   ,FSBB   ,FSBR   ,FSBRI  ,FSBRM  ,FSBRB
                                ,RED    ,RED    ,RMW    ,RMW    ,RED    ,IMM    ,RMW    ,RMW
                                ;
                                ;FMP    ,FMPL   ,FMPM   ,FMPB   ,FMPR   ,FMPRI  ,FMPRM  ,FMPRB
                                ,RED    ,RED    ,RMW    ,RMW    ,RED    ,IMM    ,RMW    ,RMW
                                ;
                                ;FDV    ,FDVL   ,FDVM   ,FDVB   ,FDVR   ,FDVRI  ,FDVRM  ,FDVRB
                                ,RED    ,RED    ,RMW    ,RMW    ,RMW    ,IMM    ,RMW    ,RMW
                                ;
                                ;MOVE   ,MOVEI  ,MOVEM  ,MOVES  ,MOVS   ,MOVSI  ,MOVSM  ,MOVSS
                                ,RED    ,IMM    ,WRT    ,RMW    ,RED    ,IMM    ,WRT    ,RMW
                                ;
                                ;MOVN   ,MOVNI  ,MOVNM  ,MOVNS  ,MOVM   ,MOVMI  ,MOVMM  ,MOVMS
                                ,RED    ,IMM    ,WRT    ,RMW    ,RED    ,IMM    ,WRT    ,RMW
                                ;
                                ;IMUL   ,IMULI  ,IMULM  ,IMULB  ,MUL    ,MULI   ,MULM   ,MULB
                                ,RED    ,IMM    ,RMW    ,RMW    ,RED    ,IMM    ,RMW    ,RMW
                                ;
                                ;IDIV   ,IDIVI  ,IDIVM  ,IDIVB  ,DIV    ,DIVI   ,DIVM   ,DIVB
                                ,RED    ,IMM    ,RMW    ,RMW    ,RED    ,IMM    ,RMW    ,RMW
                                ;
                                ;ASH    ,ROT    ,LSH    ,JFFO   ,ASHC   ,ROTC   ,LSHC   ,N'EXISTE PAS
                                ,IMM    ,IMM    ,IMM    ,BRK    ,IMM    ,IMM    ,IMM    ,NXI
                                ;
                                ;EXCH   ,BLT    ,AOBJP  ,AOBJN  ,JRST   ,JFCL   ,XCT    ,N'EXISTE PAS
                                ,RMW    ,PTR    ,BRK    ,BRK    ,BRK    ,BRK    ,BRK    ,NXI
                                ;
                                ;PUSHJ  ,PUSH   ,POP    ,POPJ   ,JSR    ,JSP    ,JSA    ,JRA
                                ,PSJ    ,PSH    ,POP    ,PPJ    ,BRK    ,BRK    ,BRK    ,BRK
                                ;
                                ;ADD    ,ADDI   ,ADDM   ,ADDB   ,SUB    ,SUBI   ,SUBM   ,SUBB
                                ,RED    ,IMM    ,RMW    ,RMW    ,RED    ,IMM    ,RMW    ,RMW
```

```
;CAI    ,CAIL   ,CAIE   ,CAILE  ,CAIA   ,CAIGE  ,CAIN   ,CAIG
,NOP    ,IMM    ,IMM    ,IMM    ,IMM    ,IMM    ,IMM    ,IMM
;
;CAM    CAML    ,CAMLE  ,CAMLE  ,CAMA   ,CAMGE  ,CAMN   ,CAMG
,NOP    ,RED    ,RED    ,RED    ,RED    ,RED    ,RED    ,RED
;
;JUMP   ,JUMPL  ,JUMPE  ,JUMPLE ,JUMPA  ,JUMPGE ,JUMPN  ,JUMPG
,BRK    ,BRK    ,BRK    ,BRK    ,BRK    ,BRK    ,BRK    ,BRK
;
;SKIP   ,SKIPL  ,SKIPE  ,SKIPLE ,SKIPA  ,SKIPGE ,SKIPN  ,SKIPG
,NOP    ,RED    ,RED    ,RED    ,RED    ,RED    ,RED    ,RED
;
;AOJ    ,AOJL   ,AOJE   ,AOJLE  ,AOJA   ,AOJGE  ,AOJN   ,AOJG
,IMM    ,BRK    ,BRK    ,BRK    ,BRK    ,BRK    ,BRK    ,BRK
;
;AOS    ,AOSL   ,AOSE   ,AOSLE  ,AOSA   ,AOSGE  ,AOSN   ,AOSG
,RMW    ,RMW    ,RMW    ,RMW    ,RMW    ,RMW    ,RMW    ,RMW
;
;SOJ    ,SOJL   ,SOJE   ,SOJLE  ,SOJA   ,SOJGE  ,SOJN   ,SOJG
,IMM    ,BRK    ,BRK    ,BRK    ,BRK    ,BRK    ,BRK    ,BRK
;
;SOS    ,SOSL   ,SOSE   ,SOSLE  ,SOSA   ,SOSGE  ,SOSN   ,SOSG
,RMW    ,RMW    ,RMW    ,RMW    ,RMW    ,RMW    ,RMW    ,RMW
;
;SETZ   ,SETZI  ,SETZM  ,SETZB  ,AND    ,ANDI   ,ANDM   ,ANDB
,IMM    ,IMM    ,WRT    ,WRT    ,RED    ,IMM    ,RMW    ,RMW
;
;ANDCA  ,ANDCAI ,ANDCAM ,ANDCAB ,SETM   ,SETMI  ,SETMM  ,SETMB
,RED    ,IMM    ,RMW    ,RMW    ,RED    ,IMM    ,NOP    ,RED
;
;ANDCM  ,ANDCMI ,ANDCMM ,ANDCMB ,SETA   ,SETAI  ,SETAM  ,SETAB
,RED    ,IMM    ,RMW    ,RMW    ,NOP    ,NOP    ,WRT    ,WRT
;
;XOR    ,XORI   ,XORM   ,XORB   ,IOR    ,IORI   ,IORM   ,IORB
,RED    ,IMM    ,RMW    ,RMW    ,RED    ,IMM    ,RMW    ,RMW
;
;ANDCB  ,ANDCBI ,ANDCBM ,ANDCBB ,EQV    ,EQVI   ,EQVM   ,EQVB
,RED    ,IMM    ,RMW    ,RMW    ,RED    ,IMM    ,RMW    ,RMW
;
;SETCA  ,SETCAI ,SETCAM ,SETCAB ,ORCA   ,ORCAI  ,ORCAM  ,ORCAB
,IMM    ,IMM    ,WRT    ,WRT    ,RED    ,IMM    ,RMW    ,RMW
;
;SETCM  ,SETCMI ,SETCMM ,SETCMB ,ORCM   ,ORCMI  ,ORCMM  ,ORCMB
,RED    ,IMM    ,RMW    ,RMW    ,RED    ,IMM    ,RMW    ,RMW
;
;ORCB   ,ORCBI  ,ORCBM  ,ORCBB  ,SETO   SETOI   ,SETOM  ,SETOB
,RED    ,IMM    ,RMW    ,RMW    ,IMM    ,IMM    ,WRT    ,WRT
;
;HLL    ,HLLI   ,HLLM   ,HLLS   ,HRL    ,HRLI   ,HRLM   ,HRLS
,RED    ,NOP    ,RMW    ,RMW    ,RED    ,IMM    ,RMW    ,RMW
;
;HLLZ   HLLZI   ,HLLZM  ,HLLZS  ,HRLZ   ,HRLZI  ,HRLZM  ,HRLZS
,RED    ,NOP    ,WRT    ,RMW    ,RED    ,IMM    ,WRT    ,RMW
;
;HLLO   HLLOM   ,HLLOM  ,HLLOS  ,HRLO   ,HRLOI  ,HRLOM  ,HRLOS
,RED    ,NOP    ,WRT    ,RMW    ,RED    ,IMM    ,WRT    ,RMW
;
;HLLE   ,HLLEI  ,HLLEM  ,HLLES  ,HRLE   ,HRLEI  ,HRLEM  ,HRLES
,RED    ,NOP    ,WRT    ,RMW    ,RED    ,IMM    ,WRT    ,RMW
;
;HRR    ,HRRI   ,HRRM   ,HRRS   ,HLR    ,HLRI   ,HLRM   ,HLRS
,RED    ,IMM    ,RMW    ,RMW    ,RED    ,NOP    ,RMW    ,RMW
;
;HRRZ   ,HRRZI  ,HRRZM  ,HRRZS  ,HLRZ   ,HLRZI  ,HLRZM  ,HLRZS
,RED    ,IMM    ,WRT    ,RMW    ,RED    ,NOP    ,WRT    ,RMW
;
;HRRO   ,HRROI  ,HRROM  ,HRROS  ,HLRO   ,HLROI  ,HLROM  ,HLROS
,RED    ,IMM    ,WRT    ,RMW    ,RED    ,NOP    ,WRT    ,RMW
;
;HRRE   ,HRREI  ,HRREM  ,HRRES  ,HLRE   ,HLREI  ,HLREM  ,HLRES
,RED    ,IMM    ,WRT    ,RMW    ,RED    ,NOP    ,WRT    ,RMW
;
;TRNN   ,TLN    ,TRNE   ,TLNE   ,TRNA   ,TLNA   ,TRNN   ,TLNN
,NOP    ,NOP    ,IMM    ,IMM    ,NOP    ,NOP    ,IMM    ,IMM
;
;TDN    ,TSN    ,TDNE   ,TSNE   ,TDNA   ,TSNA   ,TDNN   ,TSNN
,NOP    ,NOP    ,RED    ,RED    ,NOP    ,NOP    ,RED    ,RED
;
;TRZ    ,TLZ    ,TRZE   ,TLZE   ,TRZA   ,TLZA   ,TRZN   ,TLZN
,IMM    ,IMM    ,IMM    ,IMM    ,IMM    ,IMM    ,IMM    ,IMM
;
;TDZ    ,TSZ    ,TDZE   ,TSZE   ,TDZA   ,TSZA   ,TDZN   ,TSZN
,RED    ,RED    ,RED    ,RED    ,RED    ,RED    ,RED    ,RED
;
;TRC    ,TLC    ,TRCE   ,TLCE   ,TRCA   ,TLCA   ,TRCN   ,TLCN
,IMM    ,IMM    ,IMM    ,IMM    ,IMM    ,IMM    ,IMM    ,IMM
;
;TDC    ,TSC    ,TDCE   ,TSCE   ,TDCA   ,TSCA   ,TDCN   ,TSCN
,RED    ,RED    ,RED    ,RED    ,RED    ,RED    ,RED    ,RED
;
;TRO    ,TLO    ,TROE   ,TLOE   ,TROA   ,TLOA   ,TRON   ,TLON
,IMM    ,IMM    ,IMM    ,IMM    ,IMM    ,IMM    ,IMM    ,IMM
;
;TDO    TSO     ,TDOE   ,TSOE   ,TDOA   ,TSOA   ,TDON   ,TSON
,RED    ,RED    ,RED    ,RED    ,RED    ,RED    ,RED    ,RED
```

```
UCPROG - UNITE.CENTRALE.PROGRAMMEE POUR TRAVAIL EN PAGE 09.08.70        MACRO.V43 23:10  2-SEP-70 PAGE 8
UCPROG.MAC      TABLEAU DE DEFINITION DES GENRES D'INSTRUCTIONS

                                >+IRP
000470' 260570  414412          EXP     ZZZ
000471' 265532  655326          EXP     ZZZ
000472' 265532  655326          EXP     ZZZ
000473' 265531  645024          EXP     ZZZ
000474' 245122  434722          EXP     ZZZ
000475' 223502  244716          EXP     ZZZ
000476' 224472  045116          EXP     ZZZ
000477' 164511  641122          EXP     ZZZ
000500' 163512  245022          EXP     ZZZ
000501' 223500  044720          EXP     ZZZ
000502' 004472  001116          EXP     ZZZ
000503' 200111  641122          EXP     ZZZ
000504' 164112  235022          EXP     ZZZ
000505' 223502  245020          EXP     ZZZ
000506' 206102  041322          EXP     ZZZ
000507' 326143  061426          EXP     ZZZ
000510' 021033  661430          EXP     ZZZ
000511' 303502  244720          EXP     ZZZ
000512' 224562  041020          EXP     ZZZ
000513' 204103  434716          EXP     ZZZ
000514' 163471  661430          EXP     ZZZ
000515' 306143  061616          EXP     ZZZ
000516' 163471  634720          EXP     ZZZ
000517' 306143  061430          EXP     ZZZ
000520' 224512  245122          EXP     ZZZ
000521' 224143  061430          EXP     ZZZ
000522' 306112  245122          EXP     ZZZ
000523' 224512  040000          EXP     ZZZ
000524' 164112  235022          EXP     ZZZ
000525' 223503  434720          EXP     ZZZ
000526' 224563  400016          EXP     ZZZ
000527' 204511  641122          EXP     ZZZ
000530' 164112  235022          EXP     ZZZ
000531' 224100  000720          EXP     ZZZ
000532' 224472  045116          EXP     ZZZ
000533' 204511  641122          EXP     ZZZ
000534' 204000  035622          EXP     ZZZ
000535' 223502  244734          EXP     ZZZ
000536' 004472  001116          EXP     ZZZ
000537' 340111  640022          EXP     ZZZ
000540' 167002  235000          EXP     ZZZ
000541' 223502  244734          EXP     ZZZ
000542' 224472  001116          EXP     ZZZ
000543' 340111  640022          EXP     ZZZ
000544' 167002  235000          EXP     ZZZ
000545' 223560  045634          EXP     ZZZ
000546' 204163  441034          EXP     ZZZ
000547' 343473  470716          EXP     ZZZ
000550' 204102  041020          EXP     ZZZ
000551' 203471  634716          EXP     ZZZ
000552' 163502  041020          EXP     ZZZ
000553' 204101  634716          EXP     ZZZ
000554' 163471  641020          EXP     ZZZ
000555' 204102  040716          EXP     ZZZ
000556' 163471  634700  IFN ZZ-1,<EXP   ZZZ>          ;POUR LE RESTANT EVENTUEL

000557'                         LIT
000557' 201040  777777
000560' 263740  000000
000561' 201040  777776
000562' 263740  000000
000563' 260740  000026'
000564' 254000  000101'
000565'                         VAR
                                END

NO ERRORS DETECTED

PROGRAM BREAK IS 000565


UCPROG - UNITE.CENTRALE.PROGRAMMEE POUR TRAVAIL EN PAGE 09.08.70        MACRO.V43 23:10  2-SEP-70 PAGE 9
UCPROG.MAC      SYMBOL TABLE

ACCES0          000413'         ACCES1          000417'         ACCES2          000425'         ACCESM          000405'
ACCMEM          000417' EXT     ACCMM1          000303' EXT     ACCPAG          000460' EXT     ACCUS           000452' EXT
ACMMAB          000365' EXT     ACU             000010  SPD     ADR             000004  SPD     ALAPRB          000003  SPD
APRDRP          000311' EXT     ASCIIL          000001  SPD     BKSHFT  777777  777771  SPD     BLKSIZ          000200  SPD
BLTINB          000366'         BRK             000014  SPD     BRKDSP          000216'         BYT             000012  SPD
BYTDSP          000153'         CHN             000006  SPD     CORE            000011  SPD     DAT             000005  SPD
DATABS          000054' EXT     DATREL          000230' EXT     DDTOUT          000003  SPD     DEVCWR          000001  SPD
DISPAT          000112'         DRP             000000  SPD     DUMP            000017  SPD     EXIT            000012  SPD
FORTY           000403' EXT     HGHOFF          000002  SPD     HGHON           000003  SPD     HIGHSW  777777  777775  SPD
HORPAG          000004  SPD     IDX             000011  SPD     ILEGAB          000316' EXT     ILLINS          000110' EXT
ILM             020000  SPD     ILMENB          000016  SPD     IMM             000010  SPD     IMMB            000001  SPD
IMMDSP          000147'         INS             000016  SPD     INSPTR          000461'         INSTAB          000470'
JDABLK          000001  SPD     JOBCNI          000312' EXT     JOBPAG          000140  SPD     JOBTPC          000006' EXT
JRS             000006  SPD     JRSDSP          000254'         JSP             000005  SPD     JSPDSP          000252'
JSR             000004  SPD     JSRDSP          000245'         LOALOC          400000  SPD     LOC             000007  SPD
LSTCB   777777  777777  SPD     MARGE           000002  SPD     MEMORY          000008  SPD     METABS          000457' EXT
NADRB           000002  SPD     NOP             000016  SPD     NOPDSP          000243'         NORPAG          000010  SPD
NXI             000013  SPD     NXIDSP          000207'         NXIMES          000211'         OPTESS  777777  777777  SPD
PADRDX          000425' EXT     PAGBLN          000013' EXT     PAGADR          000441' EXT     PAGEPC          000346' EXT
PAGLOC          002000  SPD     PDLOVF          000310'         PEEK            000033  SPD     PIL             000017  SPD
PILMES          000317'         PINSAC          000177' EXT     PINSCD          000042' EXT     PINSDX          000032' EXT
POP             000003  SPD     POPDSP          000322'         PPWDS1          000335'         PPJ             000017  SPD
PPJDSP          000337'         PSW             000002  SPD     PSWDSP          000277'         PSJ             000001  SPD
PSJDSP          000260'         PTR             000015  SPD     PTRDSP          000355'         PURARG          000008  EXT
PUROPE          000453' EXT     RED             000007  SPD     REDDSP          000132'         RMW             000011  SPD
RMWDSP          000141'         RNGMEM          000370' EXT     RNGMM1          000375' EXT     RNGMNB          000008  EXT
SD              000015  SPD     SETALB          000026'         SETALL          000027'         SETDDT          000002  SPD
SETPAG          000455'         SETRET          000052'         SLEEP           000031  SPD     TAC             000001  SPD
TAC1            000002  SPD     TAC2            000003  SPD     TTY             000008  SPD     US1APB          000022' INT
US1AP1          000000' INT     USERGO          000242' EXT     USRAPR          000105' EXT     USRSAV          000000  EXT
USRTRP          000105' INT     USRXIT          000052' EXT     UU              000005  SPD     UUO             000016  SPD
UUOOFF          000001  SPD     UUOON           000000  SPD     UUOSW   777777  777775  SPD     UUOSY1          000484' EXT
VIOMEM          000053'         VIOMM1          000070'         VIOMM3          000076'         VIOMM4          000077' INT
VIOMM5          000101'         VIOUUO          000377'         WRT             000003  SPD     WRTDSP          000134'
XCTIN1          000444'         XCTINS          000443'         XCTOPE          000454' EXT     ZZ              000007  SPD
ZZZ     163471  634700  SPD

3K CORE USED
```

ACCES0#	449#	472												
ACCES1	170	451	453#											
ACCES2	275	459#												
ACCESM	165	233	245	254	265	358	382	410	431	443#	491			
ACCMEM	81#	453												
ACCMM1	81#	362												
ACCPAG	79#	82#	121	493										
ACCUS	80#	135	282	285	299	332	340	343	344	360	364	377	384	388
	393	401	405	412	413	461	480	481	483					
ACMMAB	81#	131	345	378	394	414	417							
ACU	17#	138	282	285	332	340	343	344	360	364	377	384	388	393
	401	405	412	413	416	421	424							
ADR	13#	109	110	111	112	116	118	119	120	129	130	136	137	156
	172	246	248	255	266	267	270	271	272	274	276	277	278	281
	286	297	329	334	335	344	350	352	353	360	377	381	393	398
	400	411	412	416	419	421	424	432	444	445	448	455	458	462
	463	465												
ALAPRB	67#	89#	155	175										
APRORP	79#	102	368											
ASCIIL	47#													
BKSHFT	58#													
BLKSIZ	57#													
BLTIN0	418#	426												
BRK	225#	691	692	692#	693#	694	697#	698	698#	700	700#	702#	703	
BRKOSP	224	296#												
BYT	223#	680#	681	681#										
BYTOSP	222	260#												
CHN	15#													
CORE	36#													
DAT	14#	122	153	305										
DATABS	83#	154												
DATREL	79#	123	306											
DDTOUT	33#													
DEVCHR	34#													
DISPAT	176	177	211#	213	214	215	216	217	218	219	220	221	222	223
	224	225	226	227	228									
DR#	9#	155	163	164	166	168	169	171	175	244	253	264	380	409
	430	449	454	490										
DUMP	50#													
EXIT	37#													
FORTY	82#	434												
HGHOFF	30#													
HGHON	31#													
HIGHSW	27#													
HORPAG	68#	90#	163	166	171									
IDX	18#	133	134	135	136	140	162	176	238	359	363	411	419	445
	447	448	450	456	459	460	461	462	464	465				
ILEGAB	79#	291	373											
ILLINS	79#	209												
ILM	40#	103	107											
ILMENB	38#													
IMM	221#	680#	682#	683#	684#	685#	686#	687#	688	688#	689	689#	690#	691
	691#	694#	695#	696	696#	699#	702	704#	705	705#	706#	708	708#	709
	709#	710	710#	711#	712	712#	713	714#	715#	716#	717#	718#	719#	720#
	721#	723	723#	725	725#	726	727#	728	728#	729#	730	730#		
IMMB	65#	67	87#	89	164	169	253	430	449	490				
IMMOSP	220	253#												
INS	21#	132	137	139	255	262	268	269	281	283	284	297	298	300
	301	303	304	307	308	350	381	432	433	434	476	477	478	479
	480	481	483	484										
INSPTR	150	497#												
INSTAB	499	501	503	505	507	509	511	531#						
JOABLK	56#													
JOBCNI	82#	104	369											
JOBPAG	62#													
JOBTPC	82#	109												
JRS	219#													
JRSOSP	218	333#												
JSP	218#	677#												
JSPOSP	217	331#												
JSR	217#	677#												
JSROSP	216	326#												
LOALOC	55#	118	158											
LOC	16#	132	267	271	278	321	328	348	359	363	379	383	395	415
	418	423	455	458	467	468	469	476						
LSTCR	5	96#												
MARGE	92#	115												
MEMORY	49#													
METABS	81#	361	399	443	446	470	492							
NADRR	66#	67	88#	89	168	244	253	264	380	409	454			
NOP	227#	695#	696#	698#	706#	707#	713#	714#	716	717	718#	720	721	722#
	723#	724	724#											
NOPOSP	226	321#												
NORPAG	69#													
NXI	224#	677	678	678#	679	679#	680	680#	691#	692#				
NXIOSP	223	290#												
NXIMES	290	293#												
OPTESS	70	106												
PADROX	81#	82#	459											
PAGOLN	80#	112	114											
PAGADR	82#	119	466	469	471									
PAGEPC	80#	110	120	122	130	153	156	159	172	184	261	263	311	316
	327	329	331	335	346	353	397	400						
PAGLOC	54#													
POLOVF	342	367#	386	403										
PEEK	42#													
PIL	22#	121	123	131	144	147	151	154	157	163	170	173	174	177
	233	239	240	241	245	246	247	248	254	265	266	272	273	274
	275	276	286	296	306	309	326	330	333	336	345	349	351	354
	358	361	362	378	379	382	383	394	399	406	410	414	415	417
	418	422	423	425	431	443	444	446	447	452	453	457	466	470
	471	491	492											
PILMES	372	374#												
PINSAC	80#	138	280											
PINSCD	80#	141												
PINSDX	80#	133												
POP	216#	677#	693#											
POPOSP	215	377#												
PPWOS1	366	388#												
PPJ	228#	677#	693#											
PPJOSP	227	393#												
PSW	215#	677#	693											
PSWOSP	214	358#												
PSJ	214#	677	693											
PSJOSP	213	340#												
PTR	226#	692												
PTROSP	225	409#												

PURARG	81#													
PUROPE	80#	304	305	314	315	478	484							
RED	220#	680#	681#	682	682#	683#	684	684#	685	686	686#	687#	688#	689
	689#	690	694	694#	696#	697	697#	698#	699	699#	705	705#	706	706#
	707#	708#	709	709#	710#	711#	712#	713#	714	714#	715#	716#	717	717#
	718	718#	719#	720#	721	721#	722	724	724#	726	726#	727	728#	729
	729#	730#	731	731#										
REDDSP	219	233#												
RMW	222#	680#	681#	682	682#	683	683#	684	684#	685#	686	686#	687	687#
	688#	689#	690	690#	691#	694#	695	701	701#	702	703#	704	704#	705#
	706	707	708	708#	709#	710	711	711#	712	712#	713#	714	714#	715
	715#	716#	717#	718	718#	719	719#	720#	721#	722	722#			
RMWDSP	221	240	244#											
RNGMEM	79#	249	273	287	349	389	420							
RNGMM1	81#	422	425											
RNGMM0	81#													
SD	52#													
SETAL0	129#	369												
SETALL	130#	157	174											
SETDDT	32#													
SETPAG	296	326	333	351	490#									
SETRET	151#	173												
SLEEP	41#													
TAC	10#	11	12	102	103	104	107	114	115	116	141	142	143	145
	146	148	149	150	161	176	177	183	184	239	241	260	261	263
	279	280	282	285	299	300	301	303	311	312	313	314	315	316
	327	328	331	332	340	341	342	343	346	347	348	364	365	366
	368	370	384	385	386	387	388	395	396	397	398	481	482	483
	484	485	499	501	503	505	507	509	511					
TAC1	11#	158	158	159	160	367	369							
TAC2	12#													
TTY	46#													
US1AP0	77	117	121#	185										
US1AP1	77	102#												
USERGO	82#	124	317											
USRAPR	83#	183												
USRSAV	79#													
USRTRP	77	105	183#	371										
USRXIT	79#	151												
UU	149	228#	497	499	501	503	505	507	509	511	677	677#	678	678#
	679	679#	680	680#	681	681#	682	682#	683	683#	684	684#	685	685#
	686	686#	687	687#	688	688#	689	689#	690	690#	691	691#	692	692#
	693	693#	694	694#	695	695#	696	696#	697	697#	698	698#	699	699#
	700	700#	701	701#	702	702#	703	703#	704	704#	705	705#	706	706#
	707	707#	708	708#	709	709#	710	710#	711	711#	712	712#	713	713#
	714	714#	715	715#	716	716#	717	717#	718	718#	719	719#	720	720#
	721	721#	722	722#	723	723#	724	724#	725	725#	726	726#	727	727#
	728	728#	729	729#	730	730#	731	731#						
UUO	20#	21												
UUOOFF	29#													
UUOON	28#													
UUOSW	26#	27												
UUOSY1	80#	435												
VIOMEM	113	153#												
VIOMM1	166#													
VIOMM3	162	172#	178	242										
VIOMM4	77	167	173#											
VIOMM5	175#	310												
VIOUUO	210	430#												
WRT	161	213#	686#	687	687#	688	704#	707#	710#	713#	715	715#	716	716#
	717#	719#	720	720#	721#	722#								
WRTDSP	212	238#												
XCTIN1	256	477#												
XCTINS	234	247	322	476#										
XCTOPE	82#	482	485											
ZZ	496#	499	500	500#	501	502	502#	503	504	504#	505	506	506#	507
	508	508#	509	510	510#	511	512	512#	521#	677	677#	678#	678	679#
	679	680#	680	681#	681	682#	682	683#	683	684#	684	685#	685	686#
	686	687#	687	688#	688	689#	689	690#	690	691#	691	692#	692	693#
	693	694#	694	695#	695	696#	696	697#	697	698#	698	699#	699	700#
	700	701#	701	702#	702	703#	703	704#	704	705#	705	706#	706	707#
	707	708#	708	709#	709	710#	710	711#	711	712#	712	713#	713	714#
	714	715#	715	716#	716	717#	717	718#	718	719#	719	720#	720	721#
	721	722#	722	723#	723	724#	724	725#	725	726#	726	727#	727	728#
	728	729#	729	730#	730	731#	731							
ZZZ	677#	677	678#	678	679#	679	680#	680	681#	681	682#	682	683#	683
	684#	684	685#	685	686#	686	687#	687	688#	688	689#	689	690#	690
	691#	691	692#	692	693#	693	694#	694	695#	695	696#	696	697#	697
	698#	698	699#	699	700#	700	701#	701	702#	702	703#	703	704#	704
	705#	705	706#	706	707#	707	708#	708	709#	709	710#	710	711#	711
	712#	712	713#	713	714#	714	715#	715	716#	716	717#	717	718#	718
	719#	719	720#	720	721#	721	722#	722	723#	723	724#	724	725#	725
	726#	726	727#	727	728#	728	729#	729	730#	730	731#	731		

GENRE	187#	211												
X	205#	212	213	214	215	216	217	218	219	220	221	222	223	224
	225	226	227											
Y	514#	531												

```
OPPROG  TRAITEMENT PREALABLE DES OPERATEURS PROGRAMMES SYSTEME 09.08.78 MACRO,V43 18:54  2-SEP-78 PAGE 1
CB      .MAC        DECLARATIONS

                                ;CETTE FILE A ETE ASSEMBLEE AVEC LA FILE DES PARAMETRES COMMUNS
                                ;AUX SOUS-PROGRAMMES DE PAGINATION  CB.MAC

                                XLIST
                                LIST
                                TITLE   OPPROG  TRAITEMENT PREALABLE DES OPERATEURS PROGRAMMES SYSTEME 09.08.78
                                        SUBTTL  DECLARATIONS


                                ;DEFINITION DES ACCUMULATEURS DE UUOCON

                        000000  ORP==   0
                        000001  TAC==   1
                        000002  TAC1==  TAC+1
                        000003  TAC2==  TAC+2           ;3 ACCUMULATEURS DE TRAVAIL
                        000005  DAT==   5               ;ADRESSES DE DISPATCH...
                        000006  CHN==   6               ;NUMERO DE CANAL OU D'ACCUMULATEUR DE L'UUO
                        000016  UUO==   16              ;CONTIENT L'UUO EN COURS DE CONTROLE
                        000017  PIL==   17              ;POINTEUR DE PILE DE TRAVAIL


                                ;POSITIONS DE RANGEMENT EXTERIEURES A UUOCON

                                INTERN  UUOINT  ,USRXIT ,USERG0 ,ILLINS ,IMEMXT ,ILEGA0 ,XCTOPE ,ILEGAL
                                INTERN  MEMINI  ,TAPMES ,UUOSY1 ,XCTUUO ,UUOERR

                                EXTERN  FORTY   ,TACSAV ,PILSAV ,USR41  ,ACCUS  ,USRREL ,MODUMP ,MEMFIL ,PAGADR
                                EXTERN  DEPART  ,UUO0   ,CORUUO ,CPOPJ  ,PILPTR ,PURUUO ,PUROPE ,ACCPAG ,APRORP
                                EXTERN  ACCMEM  ,METABS ,RNGMEM ,IORELE ,STSSET ,DATABS ,STSGET ,UOADAB ,USRSAV
                                EXTERN  TIN     ,TOUT   ,TTYUUO ,UFILE  ,UINBF  ,UINIT  ,UOPEN  ,UOUTBF ,UIFILE
                                EXTERN  JOBSA   ,PAGEPC ,USRENB ,CNVCAN ,UOFILE ,CLSUUO ,RLSUUO ,PUUOAC ,PUUOPE

                                EXTERN  USRSA   ,USRFF  ,USERGO ,DATREL ,VIOMM4

  000000'                       HISEG


                                        SUBTTL  ENTREE ET ECLATEMENTS

  000000' 200700  000000* UUOINT: MOVE    UUO,FORTY       ;ICI SUR TRAPPE EN 41 D'UNE UUO
  000001' 607700  740000          TLNN    UUO,740000
  000002' 607700  077000          TLNN    UUO,077000
  000003' 254000  000023'         JRST    UUOSYS  ;UUO SYSTEME
  000004' 550200  000000*         HRRZ    ADR,USR41       ;ADRESSE DE JSR
  000005' 322200  000023'         JUMPE   ADR,UUOSYS      ;PAS D'ADRESSE
  000006' 135040  000000*         LDB     TAC,PUUOPE      ;CODE OPERATION
  000007' 302040  000016          CAIE    TAC,<<IN.>>>-9
  000010' 306040  000017          CAIN    TAC,<<OUT.>>>-9 ;IN. OU OUT. A ARGUMENT
  000011' 254000  000056'         JRST    FORINO          ;OUI
  000012' 306040  000020          CAIN    TAC,<<DATA.>>>-9
  000013' 254000  000071'         JRST    FORDAT          ;OUI - DATA. ADRESSE A METTRE EN ABSOLU
  000014' 201240  000000* FORRET: MOVEI   DAT,UUO0        ;RANGER UNE ADRESSE ABSOLUE
  000015' 260740  000000*         PUSHJ   PIL,DATARS      ;...
  000016' 260740  000000*         PUSHJ   PIL,METARS      ;PLUS SUR - MAIS CE DEVRAIT Y ETRE
  000017' 260740  000000*         PUSHJ   PIL,ACCPAG      ;POUR Y SAUTER ENSUITE
  000020' 200040  000014*         MOVE    TAC,UUO0        ;DRAPEAUX ET PC ARRETE
  000021' 202047  000000          MOVEM   TAC,(LDC)       ;EN PLACE A L'ADRESSE
  000022' 344200  000047'         AOJA    ADR,USERG0      ;SAUT A L'ADRESSE SUIVANTE
  000023'                 UUOSYS:                         ;ICI POUR LES UUOS SYSTEME
  000023' 370200  000020*         SOS     ADR,UUO0        ;COMPENSER L'EFFET DU PREMIER JSR *****
  000024' 202200  000000*         MOVEM   ADR,PAGEPC      ;COMPTEUR PROGRAMME GENERAL
  000025' 201240  000024*         MOVEI   DAT,PAGEPC      ;ADRESSE DE CE QUE L'ON MET EN ABSOLU
  000026' 260740  000015*         PUSHJ   PIL,DATARS      ;MISE EN ABSOLU DE PAGEPC
  000027' 200700  000000* UUOSY1: MOVE    UUO,FORTY       ;(UUO) I= UUO TELLE QUE TRAPPEE
  000030' 135100  000006*         LDB     TAC1,PUUOPE     ;CODE OPERATION
  000031' 306100  000777          CAIN    TAC1,777        ;CODE SPECIAL ?
  000032' 254000  000054'         JRST    UUOPIL          ;OUI...
  000033' 301100  000040          CAIL    TAC1,40         ;0 - 37 SONT ILLEGALES
  000034' 301100  000100          CAIL    TAC1,100
  000035' 254000  000235'         JRST    ILLINS          ;ILLEGALE INSTRUCTION > 100 (8)
  000036' 135300  000000*         LDB     CHN,PUUOAC      ;NUMERO DE L'ACCUMULATEUR
  000037' 241100  777777          ROT     TAC1,-1         ;POUR DISPATCH
  000040' 414242  000054'         SETM    DAT,UUOTAB-20(TAC1)
  000041' 607100  400000  DISPAT: TLNN    TAC1,400000
  000042' 204240  000005          MOVS    DAT,DAT         ;PARTIE GAUCHE DU MOT DE LA TABLE
  000043' 260745  000000          PUSHJ   PIL,(DAT)       ;DISPATCH
  000044' 334000  000000          SKIPA                   ;RETOUR ORDINAIRE
  000045' 350000  000025*         AOS     PAGEPC          ;RETOUR+1
  000046' 350200  000045* USRXIT: AOS     ADR,PAGEPC      ;PROGRESSION NORMALE D'INSTRUCTION
  000047' 260740  000017* USERG0: PUSHJ   PIL,ACCPAG      ;AVEC UNE ADRESSE ABSOLUE
  000050' 201240  000004          MOVEI   DAT,ADR         ;CONVERTIR ADR EN RELATIF
  000051' 260740  000000*         PUSHJ   PIL,DATREL      ;...
  000052' 202200  000046*         MOVEM   ADR,PAGEPC      ;POUR EN SORTIR
  000053' 254000  000000*         JRST    USERGO          ;EN ZONE BASSE

                                ;CODES SPECIAUX POUR LES INSTRUCTIONS DE PILE

  000054' 346700  000235' UUOPIL: AOJN    UUO,ILLINS      ;L'UUO DOIT ETRE -1 (VOIR MONITEUR)
  000055' 254000  000000*         JRST    VIOMM4          ;DANS USERAPR


  000056' 135240  000036* FORINO: LDB     DAT,PUUOAC      ;REGISTRE
  000057' 271240  000000*         ADDI    DAT,ACCUS
  000060' 260740  000026*         PUSHJ   PIL,DATARS
  000061' 261740  000004          PUSH    PIL,ADR
  000062' 550205  000000          HRRZ    ADR,(DAT)       ;QUANTITE MISE EN ABSOLU=ADRESSE AVANT FORMAT
  000063' 260740  000000*         PUSHJ   PIL,ACCMEM      ;CETTE POSITION
  000064' 201247  000000          MOVEI   DAT,(LDC)
  000065' 260740  000060*         PUSHJ   PIL,DATARS      ;MISE EN ABSOLU
  000066' 260740  000000*         PUSHJ   PIL,RNGMEM      ;RANGER
  000067' 262740  000004          POP     PIL,ADR
  000070' 254000  000014'         JRST    FORRET          ;RETOUR

  000071' 201240  000027* FORDAT: MOVEI   DAT,FORTY       ;CA A METTRE EN ABSOLU
  000072' 260740  000065*         PUSHJ   PIL,DATARS
  000073' 254000  000014'         JRST    FORRET          ;RETOUR
```

```
                                        SUBTTL  TABLE DE RENVOI

000074' 000114' 000000* UUOTAB: XWD     UCALL   ,UINIT  ;(40 41)        CALL    INIT
000075' 000252' 000252'         XWD     ILEGAL  ,ILEGAL ;(42 43)        UUO42   UUO43
000076' 000252' 000252'         XWD     ILEGAL  ,ILEGAL ;(44 45)        UUO44   UUO45
000077' 000252' 000126'         XWD     ILEGAL  ,UCALLI ;(46 47)        UUO46   CALLI
000100' 000000* 000000*         XWD     UOPEN   ,TTYUUO ;(50 51)        OPEN    TTCALL
000101' 000252' 000252'         XWD     ILEGAL  ,ILEGAL ;(52 53)        UUO52   UUO53
000102' 000252' 000000*         XWD     ILEGAL  ,UFILE  ;(54 55)        UUO54   RENAME
000103' 000000* 000000*         XWD     TIN     ,TOUT   ;(56 57)        INPUT   OUTPUT
000104' 000000* 000466'         XWD     STSSET  ,CHNCAN ;(60 61)        SETSTS  STATO
000105' 000000* 000466'         XWD     STSGET  ,CHNCAN ;(62 63)        GETSTS  STATZ
000106' 000000* 000000*         XWD     UINBF   ,UOUTBF ;(64 65)        INBUF   OUTBUF
000107' 000000* 000103*         XWD     TIN     ,TOUT   ;(66 67)        IN      OUT
000110' 000000* 000000*         XWD     CLSUUO  ,RLSUUO ;(70 71)        CLOSE   RELEAS
000111' 000466' 000466'         XWD     CHNCAN  ,CHNCAN ;(72 73)        MTAPE   GETF
000112' 000466' 000466'         XWD     CHNCAN  ,CHNCAN ;(74 75)        USETI   USETO
000113' 000000* 000000*         XWD     UIFILE  ,UOFILE ;(76 77)        LOOKUP  ENTER

                                        SUBTTL  OPERATEURS PROGRAMMES DU TYPE CALL (OU CALLI)

                                ;CALL   UUOS.RECHERCHE DANS LA TABLE DES CODES SYMBOLIQUES CALTAB

000114' 260740  000000* UCALL:  PUSHJ   PIL,UOADAB      ;ADRESSE DE L'APPELLATION SYMBOLIQUE
000115' 313200  000000*         CAMLE   ADR,USRREL      ;ADRESSE DE VALIDITE
000116' 254000  000233'         JRST    UUOERR
000117' 260740  000063*         PUSHJ   PIL,ACCMFM      ;POUR OBTENIR L'ARGUMENT
000120' 200247  000000          MOVE    DAT,(LOC)       ;PRENDRE LE SYMBOLE
000121' 200100  000471'         MOVE    TAC1,[XWD -UCALEN-CCLLEN-1,-CCLLEN];POUR LES 2 TABLES
000122' 312242  000152'         CAME    DAT,CALTAB(TAC1);TROUVE CE SYMBOLE?
000123' 253100  000122'         AOBJN   TAC1,.-1        ;NON-POURSUIVRE SUR TOUTE LA TABLE
000124' 542100  000016          HRRM    TAC1,UUO        ;OUI- OU BIEN AU BOUT DE LA TABLE
000125' 661700  047000          TLO     UUO,(CALLI)     ;POUR L'EXECUTION


                                ;CALLI UUOS.LE CODE NUMERIQUE EST DANS L'INSTRUCTION ELLE MEME

000126' 570100  000016  UCALLI: HRRE    TAC1,UUO        ;PARTIE ADRESSE CALCULEE (AVANT TRAPPING)
000127' 311100  000472'         CAML    TAC1,[EXP -CCLLEN];CONTROLE POUR LES CALLIS <0
000130' 301100  000046          CAIL    TAC1,UCALEN     ;CONTROLE DE LA VALEUR NUMERIQUE
000131' 263740  000000          POPJ    PIL,            ;NON VALABLE- NON-OPERATION
000132' 262757  000000          POP     PIL,(PIL)       ;DEJA EU UN PUSHJ , CORRIGER LA PILE
000133' 217000  000002          MOVMS   TAC1            ;VALEUR ABSOLUE
000134' 231100  000011          IDIVI   TAC1,<^D36/UU>  ;UTILISATION DE LA TABLE DE DISPATCH REDUITE
000135' 271100  000221'         ADDI    TAC1,CALTYP     ;DEBUT DE CETTE TABLE
000136' 602700  400000          TRNE    UUO,1B18        ; CALLI <0 ?
000137' 275100  000001          SUBI    TAC1,CALTYP-CCLTYP;OUI - CORRIGER
000140' 505100  440400          HRLI    TAC1,(POINT UU) ;FAIRE UN POINTEUR DE BYTE
000141' 133000  000002          IBP     TAC1            ;AU MOINS UNE INCREMENTATION
000142' 365140  000141'         SOJGE   TAC2,.-1        ;DECALER DANS LE MOT POUR LE BON DISPATCH
000143' 135100  000002          LDB     TAC1,TAC1       ;CHARGER L'ARGUMENT DE DISPATCH
000144' 241100  777777          ROT     TAC1,-1         ;POUR LE BON COTE DE LA SECONDE TABLE
000145' 414242  000226'         SETM    DAT,CALITA(TAC1);ADRESSES DE DISPATCH
000146' 254000  000041'         JRST    DISPAT

                                DEFINE  CALLIS,<
                                X       RESET,  RESET   ;CALLI  0       CALL    [SIXBIT /RESET /]
                                X       DDTIN,  CKACA   ;CALLI  1       CALL    [SIXBIT /DDTIN /]
                                X       SETDDT, CKACA   ;CALLI  2       CALL    [SIXBIT /SETDDT/]
                                X       DDTOUT, CKACA   ;CALLI  3       CALL    [SIXBIT /DDTOUT/]
                                X       DEVCHR, POPJC   ;CALLI  4       CALL    [SIXBIT /DEVCHR/]
                                X       DDTGT,  POPJC   ;CALLI  5       CALL    [SIXBIT /DDTGT /]
                                X       GETCHR, POPJC   ;CALLI  6       CALL    [SIXBIT /GETCHR/]
                                X       DDTRL,  POPJC   ;CALLI  7       CALL    [SIXBIT /DDTRL /]
                                X       WAIT,   CHNCN   ;CALLI  10      CALL    [SIXBIT /WAIT  /]
                                X       CORE,   CRUUO   ;CALLI  11      CALL    [SIXBIT /CORE  /]
                                X       EXIT,   CEXIT   ;CALLI  12      CALL    [SIXBIT /EXIT  /]
                                X       UTPCLR, CHNCN   ;CALLI  13      CALL    [SIXBIT /UTPCLR/]
                                X       DATE,   POPJC   ;CALLI  14      CALL    [SIXBIT /DATE  /]
                                X       LOGIN,  ILEGA   ;CALLI  15      CALL    [SIXBIT /LOGIN /]
                                X       APRENB, APREN   ;CALLI  16      CALL    [SIXBIT /APRENB/]
                                X       LOGOUT, POPJC   ;CALLI  17      CALL    [SIXBIT /LOGOUT/]
                                X       SWITCH, POPJC   ;CALLI  20      CALL    [SIXBIT /SWITCH/]
                                X       REASSI, POPJC   ;CALLI  21      CALL    [SIXBIT /REASSI/]
                                X       TIMER,  POPJC   ;CALLI  22      CALL    [SIXBIT /TIMER /]
                                X       MSTIME, POPJC   ;CALLI  23      CALL    [SIXBIT /MSTIME/]
                                X       GETPPN, POPJC   ;CALLI  24      CALL    [SIXBIT /GETPPN/]
                                X       TRPSET, ILEGA   ;CALLI  25      CALL    [SIXBIT /TRPSET/]
                                X       TRPJEN, ILEGA   ;CALLI  26      CALL    [SIXBIT /TRPJEN/]
                                X       RUNTIM, POPJC   ;CALLI  27      CALL    [SIXBIT /RUNTIM/]
                                X       PJOB,   POPJC   ;CALLI  30      CALL    [SIXBIT /PJOB  /]
                                X       SLEEP,  POPJC   ;CALLI  31      CALL    [SIXBIT /SLEEP /]
                                X       SETPOV, APREN   ;CALLI  32      CALL    [SIXBIT /SETPOV/]
                                X       PEEK,   POPJC   ;CALLI  33      CALL    [SIXBIT /PEEK  /]
                                X       GETLIN, POPJC   ;CALLI  34      CALL    [SIXBIT /GETLIN/]
                                X       RUN,    CKACA   ;CALLI  35      CALL    [SIXBIT /RUN   /]
                                X       SETUWP, SETWP   ;CALLI  36      CALL    [SIXBIT /SETUWP/]
                                X       REMAP,  ILEGA   ;CALLI  37      CALL    [SIXBIT /REMAP /]
                                X       GETSEG, ILEGA   ;CALLI  40      CALL    [SIXBIT /GETSEG/]
                                X       GETTAB, POPJC   ;CALLI  41      CALL    [SIXBIT /GETTAB/]
                                X       SPY,    ILEGA   ;CALLI  42      CALL    [SIXBIT /SPY   /]
                                X       SETNAM, POPJC   ;CALLI  43      CALL    [SIXBIT /SETNAM/]
                                X       CPOPJ,  POPJC   ;CALLI  44      EN RESERVE
                                X       CPOPJ,  POPJC   ;CALLI  45      EN RESERVE
                                >
                                DEFINE  CCALIS,<
                                X       UUOFF,  POPJC   ;CALLI  -3      CALL    [SIXBIT /UUOFF /]
                                X       CPOPJ,  POPJC   ;CALLI  -2      EN RESERVE
                                X       LIGHTS, POPJC   ;CALLI  -1      CALL    [SIXBIT /LIGHTS/]
                                >
                                DEFINE  X (A,B),<
                                        SIXBIT  /A/
                                >
000147'                         CCLTAB: CCALIS*
000147' 656557  464600                  SIXBIT  /UUOFF/
000150' 436057  603200                  SIXBIT  /CPOPJ/
000151' 545147  506463                  SIXBIT  /LIGHTS/
                        000003  CCLLEN==.-CCLTAB
```

```
                                ;UUOS DE CWEZ DIGITAL

000152'                         CALTAB: CALLIS*
000152' 624563  456400                  SIXBIT  /RESET/
000153' 444464  515600                  SIXBIT  /DDTIN/
000154' 634564  444464                  SIXBIT  /SETDDT/
000155' 444464  576564                  SIXBIT  /DDTOUT/
000156' 444566  435062                  SIXBIT  /DEVCHR/
000157' 444464  476400                  SIXBIT  /DDTGT/
000160' 474564  435062                  SIXBIT  /GETCHR/
000161' 444464  625400                  SIXBIT  /DDTRL/
000162' 674151  640000                  SIXBIT  /WAIT/
000163' 435762  450000                  SIXBIT  /CORE/
000164' 457051  640000                  SIXBIT  /EXIT/
000165' 656460  435462                  SIXBIT  /UTPCLR/
000166' 444164  450000                  SIXBIT  /DATE/
000167' 545747  515600                  SIXBIT  /LOGIN/
000170' 416062  455642                  SIXBIT  /APRENB/
000171' 545747  576564                  SIXBIT  /LOGOUT/
000172' 636751  644350                  SIXBIT  /SWITCH/
000173' 624541  636351                  SIXBIT  /REASSI/
000174' 645155  456200                  SIXBIT  /TIMER/
000175' 556304  515545                  SIXBIT  /MSTIME/
000176' 474564  606656                  SIXBIT  /GETPPN/
000177' 646260  634564                  SIXBIT  /TRPSET/
000200' 646260  524556                  SIXBIT  /TRPJEN/
000201' 626556  645155                  SIXBIT  /RUNTIM/
000202' 605257  420000                  SIXBIT  /PJOB/
000203' 635445  456000                  SIXBIT  /SLEEP/
000204' 634564  605766                  SIXBIT  /SETPOV/
000205' 604545  530000                  SIXBIT  /PEEK/
000206' 474564  545156                  SIXBIT  /GETLIN/
000207' 626556  000000                  SIXBIT  /RUN/
000210' 634564  656760                  SIXBIT  /SETUWP/
000211' 624555  416000                  SIXBIT  /REMAP/
000212' 474564  634547                  SIXBIT  /GETSEG/
000213' 474564  644142                  SIXBIT  /GETTAB/
000214' 636071  000000                  SIXBIT  /SPY/
000215' 634564  564155                  SIXBIT  /SETNAM/
000216' 436057  603200                  SIXBIT  /CPOPJ/
000217' 436057  603200                  SIXBIT  /CPOPJ/
                000046          UCALEN==.-CALTAB

                000000          POPJC== 0
                000001          CKACA== 1
                000002          RESET== 2
                000003          CRUUO== 3
                000004          ILEGA== 4
                000005          CEXIT== 5
                000006          APREN== 6
                000007          CHNCN== 7
                000010          SETWP== 10

                000004          UU==4
                000001          ZZ==1

                                DEFINE  X (A,B),<
                                        IFE     ZZ-1,<ZZZ==B>
                                        ZZZ==ZZZ!<B-<*D36-<ZZ*UU>>>
                                        IFE     ZZ*UU-*D36,<ZZ==0
                                        EXP     ZZZ
                                >
                                ZZ==ZZ+1
                                >

000220'                         CCLTYP: CCALIS*
000220' 000000  000000                  EXP     ZZZ
                000001          ZZ==1

000221'                         CALTYP: CALLIS*
000221' 102104  000007                  EXP     ZZZ
000222' 152701  060000                  EXP     ZZZ
000223' 000021  000006                  EXP     ZZZ
000224' 000141  040100                  EXP     ZZZ
000225' 000000  000000                  EXP     ZZZ

000226' 000452' 000442' CALITA: XWD     XCTUUO  ;CKACAD
000227' 000300' 000000*         XWD     CRESET  ;CORUUO
000230' 000252' 000426'         XWD     ILEGAL  ;CCEXIT
000231' 000437' 000466'         XWD     APRENB  ;CHNCAN
000232' 000275' 000000          XWD     NOSUWP,0
                000005          CALIMX==.-CALITA

                                        SUBTTL  ROUTINES D'IMPRESSION DES ERREURS

                                ;ROUTINE DE TRAITEMENT DES UUOS ILLEGALES EN PAGINATION

000233' 201000  000413' UUOERR: MOVEI   UUOMES          ;ADRESSE DE LA CHAINE ASCIZ
000234' 254000  000253'         JRST    ILEGAB          ;ROUTINE COMMUNE

000235' 554100  000016  ILLINS: HLRZ    TAC1,UUO        ;VERIFIER SI C'EST HALT
000236' 302100  254200          CAIE    TAC1,(HALT)     ; ?
000237' 254000  000252'         JRST    ILEGAL          ;NON
                                MOVEI   [ASCIZ  \
                                ? ERREUR EN JOB PAGINE
000240' 201000  000473' HALT\]
000241' 047000  000003          CALLI   DDTOUT          ;SORTIE EN MODE DDT
000242' 201000  000421'         MOVEI   ADDMES
000243' 047000  000003          CALLI   DDTOUT
000244' 550040  000052*         HRRZ    TAC,PAGEPC      ;ADRESSE
000245' 260740  000261'         PUSHJ   PIL,MESNUM
                                MOVEI   [ASCIZ  \
000246' 201000  000502' *C\]
000247' 047000  000003          CALLI   DDTOUT
000250' 047040  000012          CALLI   1,12            ;RETOUR AU MONITEUR.....
000251' 254000  000046'         JRST    USRXIT          ;SI ON TAPE CONTINUE...
```

```
OPPROG  TRAITEMENT PREALABLE DES OPERATEURS PROGRAMMES SYSTEME 09.08.70 MACRO.V43 10:54  2-SEP-70 PAGE 4
OPPROG.MAC      ROUTINES D'IMPRESSION DES ERREURS

000252' 201000  000405' ILEGAL: MOVEI   ILLMES
000253' 047000  000003  ILEGA0: CALLI   DDTOUT          ;SORTIE EN DDT DU MESSAGE
000254' 201000  000421'         MOVEI   ADDMES          ;MESSAGE "A L'ADRESSE ABSOLUE"
000255' 047000  000003          CALLI   DDTOUT
000256' 550040  000244*         HRRZ    TAC,PAGEPC      ;ADRESSE ABSOLUE DE L'INSTRUCTION
000257' 260740  000261'         PUSHJ   PIL,MESNUM      ;IMPRESSION DE L'ADRESSE ABSOLUE
000260' 254000  000427'         JRST    CEXIT0          ;ON NE CONTINUE PAS

000261' 201240  000006  MESNUM: MOVEI   DAT,6            ;6 CHIFFRES EN TOUT
000262' 634140  000003          TDZA    TAC2,TAC2       ;MISE A ZERO PREALABLE
000263' 246100  777771  MESNU1: LSHC    TAC1,-7         ;POUR FORMAT ASCII 7 BITS PAR CARACTERE
000264' 135100  000503'         LDB     TAC1,[POINT 3,TAC,35]   ;3 BITS LES PLUS FAIBLES POIDS
000265' 242040  777775          LSH     TAC,-3          ;DECALER POUR LA PROCHAINE SAISIE...
000266' 271100  000060          ADDI    TAC1,"0"        ;CONVERTIR L'OCTAL EN ASCII
000267' 367240  000263'         SOJG    DAT,MESNU1      ;A LA 6-EME FOIS TAC1 PORTE LE PLUS HAUT POIDS
000270' 246100  000034          LSHC    TAC1,+D35-+D7   ;CADRER DANS TAC2 LE DERNIER CHIFFRE
000271' 242100  000001          LSH     TAC1,1          ;CADRER A GAUCHE TAC1 SEUL MAINTENANT
000272' 201000  000002          MOVEI   TAC1
000273' 047000  000003          CALLI   3               ;SORTIE EN DDT
000274' 263740  000000          POPJ    PIL,0           ;ET RETOUR
000275' 201040  000001  NOSUWP: MOVEI   TAC,1B35        ;BIT 35
000276' 412046  000057*         ANDCAM  TAC,ACCUS(CHN)  ;A ZERO DANS L'AC PRECISE
000277' 263740  000000          POPJ    PIL,0           ;RETOUR D'ERREUR DE CALL SETUWP

000300' 403100  000000* CRESET: SETZB   TAC1,APRORP      ;DRAPEAUX AUTORISES
000301' 070000  000000          CLOSE   MEMORY,0        ;POUR NE PAS DETRUIRE LA NOUVELLE VERSION
000302' 554040  000000*         HLRZ    TAC,USRSA       ;POUR L'UTILISATEUR...
000303' 202040  000000*         MOVEM   TAC,USRFF       ;REMISE A JOUR DE LA FF POSITION
000304' 047000  000000          CALLI   0               ;EXECUTER RESET
000305' 047100  000036          CALLI   TAC1,<SETUWP==36>;POUR ECRIRE EN SEGMENT HAUT (ESSAIS)
000306' 255000  000000          JFCL            ;TANT PIS SI CA ECHOUA
000307'                         MEMINI:
000307' 050000  000000* IMEMXT: OPEN    MEMORY,MODUMP   ;RE-INITIALISER LE CANAL
000310' 254000  000317'         JRST    OPNERR          ;PAS REUSSI!
000311' 402000  000003*         SETZM   MEMFIL+3        ;NUMERO PROJET-PROGRAMMEUR COURANT
000312' 076000  000000*         LOOKUP  MEMORY,MEMFIL   ;RE-INITIALISER LA FILE
000313' 254000  000321'         JRST    LKPERR          ;PAS REUSSI
000314' 077000  000312*         ENTER   MEMORY,MEMFIL   ;RE-INITIALISER LA FILE DE SORTIE
000315' 254000  000333'         JRST    ENTERR          ;PAS REUSSI
000316' 263740  000000          POPJ    PIL,0           ;TOUT A BIEN FONCTIONNE...

000317' 201000  000344' OPNERR: MOVEI   OPNMES          ;ADRESSE DU MESSAGE
000320' 254000  000341'         JRST    TAPMES          ;A ENVOYER AVANT DE RECOMMENCER
000321' 201040  000007  LKPERR: MOVEI   TAC,7            ;MASQUE POUR CODE D'ERREUR
000322' 201000  000000          MOVEI   0
000323' 404040  000001*         AND     TAC,MEMFIL+1    ;SAISIR LE CODE D'ERREUR
000324' 306040  000000          CAIN    TAC,0           ;NON-EXISTENCE DE LA FILE ?
000325' 201000  000357'         MOVEI   LK0MES          ;OUI...
000326' 306040  000001          CAIN    TAC,1           ;ERREUR SUR PROJET-PROGRAMMEUR ?
000327' 201000  000362'         MOVEI   LK1MES          ;OUI?..
000330' 306040  000002          CAIN    TAC,2           ;PROTECTION CONTRE LA LECTURE ?
000331' 201000  000367'         MOVEI   LK2MES          ;OUI...
000332' 254000  000341'         JRST    TAPMES          ;TAPER LE MESSAGE ADEQUAT

000333' 201040  000007  ENTERR: MOVEI   TAC,7            ;MASQUE POUR CODE D'ERREUR
000334' 404040  000001*         AND     TAC,MEMFIL+1    ;SAISIR CE CODE D'ERREUR
000335' 306040  000002          CAIN    TAC,2           ;PROTECTION CONTRE L'ECRITURE ?
000336' 201000  000373'         MOVEI   EN2MES          ;OUI...
000337' 306040  000003          CAIN    TAC,3           ;FILE EN MODIFICATION ?
000340' 201000  000400'         MOVEI   EN3MES          ;OUI...

000341' 332000  000000  TAPMES: SKIPE                   ;PRECAUTION
000342' 047000  000003          CALLI   3               ;SORTIE EN DDT DU MESSAGE ASCIZ POINTE PAR L'AC 0
000343' 254000  000000*         JRST    DEPART          ;CE QUE L'ON PEUT FAIRE DE MOINS MAL

000344' 376410  151500  OPNMES: ASCIZ   ".?PAS D'ACCES AU PERIPHERIQUE INDIQUE A LA COMMANDE.
000345' 421170  141606
000346' 426464  040652
000347' 202410  551222
000350' 502210  551222
000351' 506530  520222
000352' 472111  150652
000353' 425010  120230
000354' 405010  347632
000355' 466031  642212
000356' 000000  000000
000357' 376410  151500  LK0MES: ASCIZ   ".?PAS C'TE FILE.
000360' 415172  442500
000361' 432231  442400
000362' 376131  647252  LK1MES: ASCIZ   ".?ENNUIS AVEC [PJ,PG].
000363' 446464  040654
000364' 426064  055640
000365' 451312  043672
000366' 000000  000000
000367' 376412  247650  LK2MES: ASCIZ   ".?PROTECTION LECTURE.
000370' 426072  444636
000371' 471011  442606
000372' 522532  242400
000373' 376412  247650  EN2MES: ASCIZ   ".?PROTECTION ECRITURE.
000374' 426072  444636
000375' 471010  541644
000376' 446512  551212
000377' 000000  000000
000400' 376151  146212  EN3MES: ASCIZ   ".?FILE EN MODIFICATION.
000401' 202131  620232
000402' 476111  143222
000403' 416032  444636
000404' 470000  000000
000405' 376532  547500  ILLMES: ASCIZ   ".?UUO ILLEGALE EN PAGINATION.
000406' 446311  442616
000407' 426310  520212
000410' 471012  040616
000411' 446350  152222
000412' 476340  000000
```

```
OPPROG  TRAITEMENT PREALABLE DES OPERATEURS PROGRAMMES SYSTEME 09.08.78 MACRO.V43 10:54  2-SEP-78 PAGE 5
OPPROG.MAC      ROUTINES D'IMPRESSION DES ERREURS

  396  000413' 376030  451212  UUOMES: ASCIZ   ?ADRESSE TROP GRANDE - UUO.
       000414' 516470  520250
       000415' 512372  020216
       000416' 512031  642212
       000417' 201324  052652
       000420' 474000  000000
  397  000421' 202024  046116  ADDMES: ASCIZ   . A L'ADRESSE ABSOLUE : ?
       000422' 406112  242646
       000423' 516124  040604
       000424' 516371  452612
       000425' 201644  000000
  398                                  SUBTTL  CAS PARTICULIERS ET ROUTINE D'EXECUTION
  399
  400  000426' 326300  000452' CCEXIT: JUMPN   CHN,XCTUUO        ;SIMPLE RETOUR AU MODE MONITEUR
  401  000427' 201300  000001  CEXIT0: MOVEI   CHN,1             ;RELEASE TOUS LES CANAUX
  402  000430' 260740  000000* CEXIT1: PUSHJ   PIL,IOREIE        ;...
  403  000431' 307300  000016          CAIG    CHN,16            ;ESSAYE LES 17(8) ?
  404  000432' 344300  000430'         AOJA    CHN,CEXIT1        ;NON...
  405  000433' 201000  000343*         MOVEI   DEPART            ;REMETTRE L'ADRESSE DE DEPART DU
  406  000434' 202000  000000*         MOVEM   JOBSA             ;CONTROLEUR DE PAGINATION.
  407  000435' 070000  000000          CLOSE   MEMORY,0          ;LIQUIDER L'EXTENSION MEMOIRE
  408  000436' 047000  000012          CALLI   12                ;ET EXIT PUISQU'ON L'A VOULU
  409
  410  000437' 200046  000276* APRENB: MOVE    TAC,ACCUS(CHN)    ;DRAPEAUX A TESTER AVANT RENVOI EN CAS
  411  000440' 552040  000300*         HRRZM   TAC,APRDRP        ;D'INTERRUPTION,SERVIRA A LA GESTION DES
  412  000441' 254000  000452'         JRST    XCTUUO            ;INTERRUPTIONS...
  413
  414                                 ;CONTROLE D'ADRESSE DE L'ARGUMENT POINTE PAR L'ACCUMULATEUR UTILISE PAR L'UUO
  415
  416  000442' 550206  000437* CKACAD: HRRZ    ADR,ACCUS(CHN)    ;L'ADRESSE
  417  000443' 260740  000016*         PUSHJ   PIL,METARS        ;EN ABSOLU MAINTENANT
  418  000444' 260740  000117*         PUSHJ   PIL,ACCMEM        ;ACCES MEMOIRE . TROUVER L'ADRESSE LOCALE
  419  000445' 542346  000442*         HRRM    LOC,ACCUS(CHN)    ;ADRESSE LOCALE DANS L'AC DE TRAVAIL
  420  000446' 261740  000004          PUSH    PIL,ADR           ;SAUVER POUR RNGMEM
  421  000447' 260740  000452'         PUSHJ   PIL,XCTUUO        ;EXECUTER L'UUO NORMALEMENT
  422  000450' 262740  000004          POP     PIL,ADR           ;RESTAURER
  423  000451' 254000  000066*         JRST    RNGMEM            ;RANGER LES ARGUMENTS ET EXIT
  424
  425                                 ;ROUTINE D'EXECUTION DES UUOS APRES LES TRAVAUX PREALABLES DE CONTROLE
  426
  427  000452' 202700  000000* XCTUUO: MOVEM   UUO,PUROPE        ;PURE OPERATION
  428  000453' 202740  000000* XCTOPE: MOVEM   PIL,PILSAV        ;SAUVER LE POINTEUR DE PILE
  429  000454' 205740  000445*         MOVSI   17,ACCUS          ;RESTAURER LES AC'S
  430  000455' 251740  000017          BLT     17,17             ;...
  431  000456' 256000  000452*         XCT     PUROPE            ;EXECUTER L'UUO
  432  000457' 254000  000461'         JRST    .+2               ;RETOUR NORMAL
  433  000460' 350000  000256*         AOS     PAGEPC            ;RETOUR .+1
  434  000461' 202740  000017*         MOVEM   17,ACCUS+17       ;RESAUVER LES AC'S
  435  000462' 201740  000454*         MOVEI   17,ACCUS          ;...
  436  000463' 251740  000016*         BLT     17,ACCUS+16       ;...
  437  000464' 200740  000453*         MOVE    PIL,PILSAV        ;RECUPERER LE POINTEUR DE PILE
  438  000465' 263740  000000          POPJ    PIL,0             ;ET RETOUR A LA SEQUENCE D'APPEL
  439
  440                                 ;POUR LES INSTRUCTIONS NECESSITANT UN NO DE CANAL
  441
  442  000466' 260740  000000* CHNCAN: PUSHJ   PIL,CNVCAN        ;CHANGER LE NO DE CANAL
  443  000467' 137300  000504'         DPB     CHN,[POINT 4,UUO,12];ET LE METTRE EN PLACE
  444  000470' 254000  000452'         JRST    XCTUUO            ;SUITE NORMALE
  445  000471'                     LIT
  446  000471' 777726  777775
  447  000472' 777777  777775
  448  000473' 064247  720212
  449  000474' 512450  552644
  450  000475' 202131  620224
  451  000476' 476044  050202
  452  000477' 436231  642432
  453  000500' 052210  146250
  454  000501' 000000  000000
  455  000502' 064253  641400
  456  000503' 000300  000001
  457  000504' 270400  000016
  458  000505'                     VAR
  459                             END

NO ERRORS DETECTED

PROGRAM BREAK IS 000505
```

```
OPPROG  TRAITEMENT PREALABLE DES OPERATEURS PROGRAMMES SYSTEME 09.08.78 MACRO.V43 18:54  2-SEP-78 PAGE 6
OPPROG.MAC      SYMBOL TABLE

ACCMEM          000444' EXT     ACCPAG          000047' EXT     ACCUS           000462' EXT     ACU             000010  SPD
ADDMES          000421'         ADR             000004  SPD     ALAPRB          000003  SPD     APRDRP          000440' EXT
APREN           000006  SPD     APRENB          000437'         ASCIIL          000001  SPD     BKSHFT  777777  777771  SPD
BLKSIZ          000200  SPD     CALIMX          000005  SPD     CALITA          000226'         CALTAB          000152'
CALTYP          000221'         CCEXIT          000426'         CCLLEN          000003  SPD     CCLTAB          000147'
CCLTYP          000220'         CEXIT           000005  SPD     CEXIT0          000427'         CEXIT1          000430'
CHN             000006  SPD     CHNCAN          000466'         CHNCN           000007  SPD     CKACA           000001  SPD
CKACAD          000442'         CLSUUO          000000  EXT     CNVCAN          000466' EXT     CORE            000011  SPD
CORUUO          000227' EXT     CPOPJ           000000  EXT     CRESET          000308'         CRUUO           000003  SPD
DAT             000005  SPD     DATABS          000072' EXT     DATREL          000051' EXT     DDTOUT          000003  SPD
DEPART          000433' EXT     DEVCHR          000001  SPD     DISPAT          000041'         DRP             000008  SPD
DUMP            000017  SPD     EN2MES          000373'         EN3MES          000400'         ENTERR          000333'
EXIT            000012  SPD     FORDAT          000071'         FORINO          000056'         FORRET          000014'
FORTY           000071' EXT     HGHOFF          000002  SPD     HGHON           000003  SPD     HIGHSW  777777  777775  SPD
HORPAG          000004  SPD     IDX             000011  SPD     ILEGA           000004  SPD     ILEGA0          000253' INT
ILEGAL          000252' INT     ILLINS          000235' INT     ILLMES          000405'         ILM             020000  SPD
ILMENB          000016  SPD     IMEMXT          000307' INT     INMB            000001  SPD     INS             000016  SPD
IORELE          000430' EXT     JOABLK          000001  SPD     JOBPAG          000140  SPD     JOBSA           000434' EXT
LK0MES          000357'         LK1MES          000362'         LK2MES          000367'         LKPERR          000321'
LOALOC          400000  SPD     LOC             000007  SPD     MEMFIL          000314' EXT     MEMINI          000307' INT
MEMORY          000000  SPD     MESNU1          000263'         MESNUM          000261'         METABS          000443' EXT
MODUMP          000307' EXT     NADRB           000002  SPD     NORPAG          000010  SPD     NOSUMP          000275'
OPNERR          000317'         OPNMES          000344'         OPTESS  777777  777777  SPD     PAGADR          000000  EXT
PAGEPC          000460' EXT     PAGLOC          002000  SPD     PEEK            000033  SPD     PIL             000017  SPD
PILPTR          000000  EXT     PILSAV          000464' EXT     POPJC           000200  SPD     PUROPE          000456' EXT
PURUUO          000000  EXT     PUUOAC          000056' EXT     PUUOPE          000038' EXT     RESET           000002  SPD
RLSUUO          000110' EXT     RNGMEM          000451' EXT     SD              000015  SPD     SETDDT          000002  SPD
SETUWP          000036  SPD     SETWP           000010  SPD     SLEEP           000031  SPD     STSGET          000000  EXT
STSSET          000000  EXT     TAC             000001  SPD     TAC1            000002  SPD     TAC2            000003  SPD
TACSAV          000000  EXT     TAPMES          000341' INT     TIN             000000  EXT     TOUT            000107' EXT
TTY             000000  SPD     TTYUUO          000100' EXT     UCALEN          000246  SPD     UCALL           000114'
UCALLI          000126'         UFILE           000102' EXT     UIFILE          000000  EXT     UINBF           000000  EXT
UINIT           000074' EXT     UOADAB          000114' EXT     UOFILE          000113' EXT     UOPEN           000000  EXT
UOUTBF          000106' EXT     USERG0          000047' INT     USERGO          000053' EXT     USR41           000004' EXT
USRENB          000000  EXT     USRFF           000303' EXT     USRREL          000115' EXT     USRSA           000302' EXT
USRSAV          000000  EXT     USRXIT          000046' INT     UU              000004  SPD     UUO             000016  SPD
UUO0            000023' EXT     UUOERR          000233' INT     UUOINT          000000' INT     UUOMES          000413'
UUOOFF          000001  SPD     UUOON           000000  SPD     UUOPIL          000054'         UUOSW   777777  777775  SPD
UUOSY1          000027' INT     UUOSYS          000023'         UUOTAB          000074'         VIOMM4          000055' EXT
XCTOPE          000453' INT     XCTUUO          000452' INT     ZZ              000003  SPD     ZZZ             000000  SPD

3K CORE USED


ACCMEM     29#    99    135    418
ACCPAG     28#    56     82
ACCUS      27#    95    345    410    416    419    429    434    435    436
ACU         5#
ADDMES    314   326    397#
ADR         5#    45     46     59     62     63     81     83     85     97     98    103    133    416
           420   422
ALAPRB      5#
APRDRP     28#   348    411
APREN     266#   288#   289#
APRENB    296    410#
ASCIIL      5#
BKSHFT      5#
BLKSIZ      5#
CALIMX    298#
CALITA    161    293#   298
CALTAB    138    219#   258
CALTYP    153    155    286#
CCEXIT    295    400#
CCLLEN    137    147    216#
CCLTAB    212#   216
CCLTYP    155    282#
CEXIT     265#   288#
CEXIT0    330    401#
CEXIT1    402#   404
CHN         5#    17#    73    345    400    401    403    404    410    416    419    443
CHNCAN    119    120    124    125    296    442#
CHNCN     267#   287#   288#
CKACA     261#   287    287#   290#
CKACAD    293    416#
CLSUUO     31#   123
CNVCAN     31#   442
CORE        5#
CORUUO     28#   294
CPOPJ      28#
CRESET    294    348#
CRUUO     263#   288
DAT         5#    16#    53     64     75     77     78     83     94     95     98    100    106    136
           138   161    332    338
DATABS     29#    54     65     96    101    107
DATREL     33#    84
DDTOUT      5#   313    315    320    325    327
DEPART     28#   387    405
DEVCHR      5#
DISPAT     76#   162
DRP         5#    12#
DUMP        5#
EN2MES    381    393#
EN3MES    383    394#
ENTERR    362    378#
EXIT        5#
FORDAT     52    106#
FORINO     50     94#
FORRET     53#   104    108
FORTY      27#    41     66    106
HGHOFF      5#
HGHON       5#
HIGHSW      5#
HORPAG      5#
IDX         5#
ILEGA     264#   288#   289#   290#
ILEGA0     24    305    325#
```

ILEGAL	24	112	113	114	116	117	295	309	324#					
ILLINS	24	72	98	307#										
ILLMES	324	395#												
ILM	5#													
ILMENB	5#													
IMEMXT	24	356#												
IMMB	5#	5												
INS	5#													
IORELE	29#	402												
JOABLK	5#													
JOBPAG	5#													
JOBSA	31#	406												
LK0MES	371	390#												
LK1MES	373	391#												
LK2MES	375	392#												
LKPERR	360	367#												
LOALOC	5#													
LOC	5#	58	100	136	419									
LSTCB	5													
MEMFIL	27#	358	359	361	369	379								
MEMINI	25	355#												
MEMORY	5#	349	356	359	361	407								
MESNU1	334#	338												
MESNUM	317	329	332#											
METABS	29#	55	417											
MODUMP	27#	356												
NADRB	5#	5												
NORPAG	5#													
NOSUWP	297	344#												
OPNERR	357	365#												
OPNMES	365	389#												
OPTESS	5													
PAGADR	27#													
PAGEPC	31#	63	64	80	81	85	316	328	433					
PAGLOC	5#													
PEEK	5#													
PIL	5#	19#	54	55	56	65	78	82	84	96	97	99	101	102
	103	107	132	135	149	150	317	329	343	346	363	402	417	418
	420	421	422	428	437	438	442							
PILPTR	28#													
PILSAV	27#	428	437											
POPJC	260#	283	283#	287#	288#	289	289#	290	290#	291	291#			
PUROPE	28#	427	431											
PURUUO	28#													
PUUOAC	31#	73	94											
PUUOPE	31#	47	67											
RESET	262#	287												
RLSUUO	31#	123												
RNGMEM	29#	102	423											
SD	5#													
SETODT	5#													
SETUWP	353#													
SETWP	268#	290#												
SLEEP	5#													
STSGET	29#	120												
STSSET	29#	119												
TAC	5#	5	13#	14	15	47	48	49	51	57	58	316	328	335
	336	344	345	350	351	367	369	370	372	374	378	379	380	382
	410	411												
TAC1	5#	14#	67	68	70	71	74	75	76	137	138	139	140	146
	147	148	151	152	153	155	156	157	159	160	161	307	328	334
	335	337	339	340	341	348	353							
TAC2	5#	15#	158	333										
TACSAV	27#													
TAPMES	25	366	376	385#										
TIN	30#	118	122											
TOUT	30#	118	122											
TTY	5#													
TTYUUO	30#	115												
UCALEN	137	148	258#											
UCALL	111	132#												
UCALLI	114	146#												
UFILE	30#	117												
UIFILE	30#	126												
UINBF	30#	121												
UINIT	30#	111												
UOADAB	29#	132												
UOFILE	31#	126												
UOPEN	30#	115												
UOUTBF	30#	121												
USERG0	24	59	82#											
USERGO	33#	86												
USR41	27#	45												
USRENB	31#													
USRFF	33#	351												
USRREL	27#	133												
USRSA	33#	350												
USRSAV	29#													
USRXIT	24	81#	322											
UU	152	156	270#	283	283#	287	287#	288	288#	289	289#	290	290#	291
	291#													
UUO	5#	5	10#	41	42	43	66	98	140	141	146	154	307	427
	443													
UUO0	28#	53	57	62										
UUOERR	25	134	304#											
UUOINT	24	41#												
UUOMES	304	396#												
UUOOFF	5#													
UUOON	5#													
UUOPIL	69	90#												
UUOSW	5#	5												
UUOSY1	25	66#												
UUOSYS	44	46	61#											
UUOTAB	75	111#												
VIOMM4	33#	91												
XCTOPE	24	428#												
XCTUUO	25	293	400	412	421	427#	444							
ZZ	271#	283	283#	284#	287	287#	288#	288	289#	289	290#	290	291#	291
ZZZ	283#	283	287#	287	288#	288	289#	289	290#	290	291#	291		
CALLIS	164#	219	286											
CCALIS	204#	212	282											
X	209#	213	214	215	220	221	222	223	224	225	226	227	228	229
	230	231	232	233	234	235	236	237	238	239	240	241	242	243
	244	245	246	247	248	249	250	251	252	253	254	255	256	257
	273#	283	283#	287	287#	288#	289#	290#	291#					

```
MEMOIRE - GESTION DU FICHIER MEMOIRE : PAGES ET FRAGMENTS  09.08.78      MACRO.V43 10:49  2-SEP-78 PAGE 1
CB    .MAC      DECLARATIONS

                                ;CETTE FILE A ETE ASSEMBLEE AVEC LA FILE DES PARAMETRES COMMUNS
                                ;AUX SOUS-PROGRAMMES DE PAGINATION  CB.MAC

                                XLIST
                                LIST
                                TITLE   MEMOIRE - GESTION DU FICHIER MEMOIRE : PAGES ET FRAGMENTS  09.08.78
                                SUBTTL  DECLARATIONS

                                ;SOUS PROGRAMME ASSEMBLE AVEC LA FILE DES PARAMETRES CB.MAC

                                ;PARAMETRES D'UUOS D'ENTREE/SORTIE

                        000004  HORPAG==4          ;INDIQUE BESOIN D'UTILISER LE SEGMENT (DONC PAS DANS LA PAGE)
                        400000  IOIMP==            400000  ;IMPOSSIBLE..DTA WRITE-LOCK
                        200000  IODEV==            200000  ;DEVICE ERREUR
                        100000  IODAT==            100000  ;DATA ERREUR
                        040000  IOBKTL==           40000   ;BLOCK TROP GRAND
                        740000  IOBAD==:IOIMP!IODEV!IODAT!IOBKTL

                                INTERN  ACCMEM  ,ACCMM1 ,RNGMEM ,RNGMN1 ,MEMSOR ,MEMGET ,STATUS ,SORSEG
                                INTERN  CHKMEM  ,RNGMM1 ,CORUUO ,CORUUB ,ANYGET ,ANYSOR ,DATABB ,DATABS
                                INTERN  ACCPAG  ,ACMMAB ,ARRSUP ,RNGMNB ,CHKFRG ,DATREL ,DATRLB ,METABS
                                INTERN  RELLOC  ,TTACAB ,UOADAB

                                EXTERN  PAGLNG  ,UTILNG ,PAGADR ,PAGZDR ,MEMBLP ,SEGADR ,SEGZDR ,SEGBLK
                                EXTERN  ABSMAX  ,SEGLOC ,MAXBLK ,ACCUS  ,SVTAC  ,SVTAC2 ,PAGBLP ,PAGBLK
                                EXTERN  ILEGAL  ,ILEGAB ,TAPMES ,CPOPJ1 ,CPOPJ  ,TPOPJ  ,JOBCNI ,USRTRP
                                EXTERN  PAGEPC  ,USRREL ,USRENB ,SEGPAG ,CORIOW ,PAGBLN

                        000200  MAXK==  ^D128

                        000200  SEGSIZ==BLKSIZ  ;LONGUEUR DU SEGMENT D'ACCES
                        400000  LOALOC==400000  ;DEBUT DE L'ADRESSAGE EN ABSOLU
                 777777 777771  SGSHFT==BKSHFT

 000000'                        HISEG



                                SUBTTL  GESTION DES FRAGMENTS

                                ;ROUTINE D'ACCES MEMOIRE POUR UN SEUL BLOC EN LECTURE SEULEMENT AU MOYEN
                                ;D'UNE ADRESSE ABSOLUE. TAC,TAC1,TAC2 SONT DETRUITS . LOC PORTE ADRESSE ABSOLUE DE PAGE
                                ;
                                ;POSITION DU MOT CHERCHE LOCALE ; ADR PEUT ETRE CORRIGE POUR ELIMINER LES TROUS.

 000000' 354000  000004  ACCMM1: AOSA    ADR             ;ENTREE AVEC INCREMENTATION POUR ACCES A
                                                         ;L'ADRESSE ABSOLUE SUIVANTE
 000001' 260740  000173' ACMMAB: PUSHJ   PIL,METABS      ;MISE EN ABSOLU PREALABLE
 000002' 260740  000050' ACCMEM: PUSHJ   PIL,CHKMEM      ;VOIR SI DEJA EN MEMOIRE
 000003' 335100  000000*         SKIPGE  TAC1,SEGBLK     ;SEGMENT COURANT MERITE RECOPIE ?
 000004' 260740  000034'         PUSHJ   PIL,SORSGB      ;OUI ...
 000005' 421340  000177          ANDCMI  LOC,SEGSIZ-1    ;ADRESSE ABSOLUE DE TETE DE SEGMENT
 000006' 202340  000000*         MOVEM   LOC,SEGADR      ;CONSERVEE
 000007' 271340  000200          ADDI    LOC,SEGSIZ      ;ADRESSE ABSOLUE DE FIN DE SEGMENT
 000010' 202340  000000*         MOVEM   LOC,SEGZDR      ;POUR FACILITER LA RECHERCHE
 000011' 201047  377600          MOVEI   TAC,LOALOC-SEGSIZ(LOC)  ;ELIMINATION DU SEUIL DE CHARGEMENT
 000012' 230040  000000*         IDIV    TAC,PAGLNG      ;CALCUL DE L'ADRESSE DE TETE DE PAGE
 000013' 220040  000012*         IMUL    TAC,PAGLNG      ;...
 000014' 271040  400000          ADDI    TAC,LOALOC
 000015' 202040  000000*         MOVEM   TAC,SEGPAG      ;SOUVENIR
 000016' 201107  377600          MOVEI   TAC1,LOALOC-SEGSIZ(LOC);ELIMINATION DU SEUIL DE CHARGEMENT
 000017' 275344  000200          SUBI    LOC,SEGSIZ(ADR) ;- DEPLACEMENT DANS LE SEGMENT
 000020' 211347  000000          MOVNI   LOC,(LOC)       ;+ DEPLACEMENT DANS LE SEGMENT
 000021' 271340  000000*         ADDI    LOC,SEGLOC      ;ADRESSE D'IMPLANTATION DU SEGMENT
 000022' 505341  000000          HRLI    LOC,(TAC)       ;ADRESSE DE PAGE
 000023' 274100  000013*         SUB     TAC1,PAGLNG     ;PAGE 0 N'EXISTE PAS
 000024' 242100  777771          LSH     TAC1,SGSHFT     ;CONVERSION PAGE EN BLOCS DE DSK
 000025' 202100  000003*         MOVEM   TAC1,SEGBLK     ;SE RAPPELER LE NO DE BLOC
 000026' 260740  000328'         PUSHJ   PIL,MEMGET      ;LIRE LE SEGMENT
 000027' 263740  000000          POPJ    PIL,0           ;RETOUR OK
 000030' 201000  000133'         MOVEI   MEMMES          ;ADRESSE DU MESSAGE D'ERREUR
 000031' 254200  000000*         HALT    ILEGAB          ;EOF (ADRESSE TROP GRANDE)

                                ;ROUTINE DE SORTIE DU SEGMENT COURANT SUR DSK (OU EN PAGE COURANTE)

 000032' 553000  000025* SORSEG: HRRZS   SEGBLK          ;NEUTRALISER LE DRAPEAU
 000033' 563000  000015*         HRROS   SEGPAG          ;ANNULATION VIS A VIS DE ACCPAG
 000034' 550040  000033* SORSGB: HRRZ    TAC,SEGPAG      ;ADRESSE DE TETE DE PAGE DE CE SEGMENT
 000035' 312040  000000*         CAME    TAC,PAGADR      ;IDENTIQUE A LA PAGE COURANTE ?
 000036' 254000  000330'         JRST    MEMSOR          ;NON - TAC1 CONTIENT LE NO DE BLOC
 000037' 402000  000010* BLTSG1: SETZM   SEGZDR           ;ANNULATION DU SEGMENT (FORCE L'USAGE DE LA PAGE)
 000040' 553000  000032* BLTSEG: HRRZS   SEGBLK           ;OUI - NEUTRALISER LE DRAPEAU
 000041' 563000  000034*         HRROS   SEGPAG          ;ANNULER CECI AUSSI VIS A VIS DE ACCPAG
 000042' 274040  000006*         SUB     TAC,SEGADR      ;-DEPLACEMENT EN PAGE
 000043' 213100  000001          MOVNS   TAC1,TAC        ;+DEPLACEMENT EN PAGE
 000044' 271100  002000          ADDI    TAC1,PAGLOC     ;ADRESSE D'IMPLANTATION DE LA PAGE
 000045' 505100  000021*         HRLI    TAC1,SEGLOC     ;ADRESSE SOURCE
 000046' 251101  002177          BLT     TAC1,PAGLOC+SEGSIZ-1(TAC);TRANSFERT EN PAGE
 000047' 263740  000000          POPJ    PIL,0           ;ET RETOUR

 000050' 201344  000000  CHKMEM: MOVEI   LOC,(ADR)        ;RECOPIE (DROITE)
 000051' 307340  000017          CAIG    LOC,17          ;ADRESSAGE D'AC ?
 000052' 271340  000000*         ADDI    LOC,ACCUS       ;OUI - TRANSLATER
 000053' 305340  002000          CAIGE   LOC,PAGLOC      ;DIRECTEMENT ACCESSIBLE ?
 000054' 254000  000100'         JRST    CKMEM2          ;OUI - TERMINE
 000055' 311340  000035*         CAML    LOC,PAGADR      ;DANS LA PAGE COURANTE ?
 000056' 311340  000000*         CAML    LOC,PAGZDR      ;... ?
 000057' 254000  000066'         JRST    CKMEM1          ;NON - VOIR SI C'EST DANS SEGMENT COURANT
 000060' 274340  000055*         SUB     LOC,PAGADR      ; DEPLACEMENT
 000061' 271340  002000          ADDI    LOC,PAGLOC      ;+ADRESSE D'IMPLANTATION
 000062' 504340  000060*         HRL     LOC,PAGADR      ;PRECISER LA PAGE
 000063' 607000  000010          TLNN    DRP,NORPAG      ;BESOIN DE METTRE EN ABSOLU ?
 000064' 661000  000004          TLO     DRP,NORPAG      ;NON - POUR L'UTILISATEUR C'EST
                                                         ;COMME EN DEHORS DE SA PAGE
 000065' 254000  000100'         JRST    CKMEM2          ;TERMINE
```

```
107  000066' 461000  000004  CKMEM1: TLO     ORP,HORPAG      ;MARQUE QUE CELA VIENT DU SEGMENT
108  000067' 311340  000042•         CAML    LOC,SEGADR      ;DANS LE SEGMENT COURANT ?
109  000070' 311340  000037•         CAML    LOC,SEGZOR      ;... ?
110  000071' 263740  000000          POPJ    PIL,0           ;NON.
111  000072' 301340  400000          CAIL    LOC,LOALOC      ;PLUS PETITE ADRESSE
112  000073' 313340  000000•         CAMLE   LOC,USRRFL      ;PLUS GRANDE ADRESSE
113  000074' 254000  000102'         JRST    ILMREF          ;A L'EXTERIEUR DE CES LIMITES
114  000075' 274340  000067•         SUB     LOC,SEGADR      ; DEPLACEMENT
115  000076' 271340  000045•         ADDI    LOC,SEGLOC      ;+ADRESSE D'IMPLANTATION
116  000077' 504340  000041•         HRL     LOC,SEGPAG      ;ADRESSE ABSOLUE DE TETE DE PAGE DU BLOC
117  000100' 262757  000000  CKMEM2: POP     PIL,(PIL)       ;DOUBLE SAUT DE SORTIE
118  000101' 263740  000000          POPJ    PIL,0           ;RETOUR
119
120                                 ;CAS DE VIOLATION DE MEMOIRE VIRTUELLE
121
122  000102' 201340  022000  ILMREF: MOVEI   LOC,102211025   ;PROTECTION MEMOIRE ET HORLOGE
123  000103' 202340  000000•         MOVEM   LOC,JOBCNI
124  000104' 200340  000000•         MOVE    LOC,USRENB      ;AUTORISES
125  000105' 602340  020000          TRNE    LOC,1022        ;DEPASSEMENT DE MEMOIRE DESIRE ?
126  000106' 254000  000060•         JRST    USRTRP          ;OUI - TRAPPE A L'UTILISATEUR
127  000107' 201000  000133'         MOVEI   MEMMES          ;NON - MESSAGE SUR LA CONSOLE
128  000110' 254000  000031•         JRST    ILEGAB          ;...
129
130                                 ;ROUTINE D'ARRONDI A LA PAGE SUPERIEURE -1
131
132  000111' 275040  400000  ARRSUP: SUBI    TAC,LOALOC      ;SOUSTRAIRE LE SEUIL DE CHARGEMENT
133  000112' 230040  000023•         IDIV    TAC,PAGLNG      ;CALCUL...
134  000113' 200100  000001          MOVE    TAC1,TAC        ;NUMERO DE PAGE DANS TAC1
135  000114' 271040  000001          ADDI    TAC,1           ;POUR PAGE SUBERIEURE
136  000115' 220040  000112•         IMUL    TAC,PAGLNG      ;ADRESSE
137  000116' 271040  400000          ADDI    TAC,LOALOC      ;REMETTRE LE SEUIL DE CHARGEMENT
138  000117' 364040  000000•         SOJA    TAC,CPOPJ       ;ET RETOUR APRES DECREMENTATION
139                                 ;ROUTINE DE RANGEMENT EN MEMOIRE D'UN BLOC OU DANS LA PAGE COURANTE
140
141  000120' 354000  000004  RNGMM1: AOSA    ADR             ;INCREMENTER SYSTEMATIQUEMENT
142  000121' 275200  000001  RNGMN1: SUBI    ADR,1           ;DECREMENTE SYSTEMATIQUEMENT
143  000122' 260740  000052' RNGMN0: PUSHJ   PIL,CHKMEM      ;VOIR SI ON ACCEDE DANS LA MEME ZONE
144  000123' 563000  000040•         HRROS   SEGBLK          ;DRAPEAU DISANT QUE CE SEGMENT DEVRA ETRE REECRIT
145  000124' 254000  000002'         JRST    ACCMEM          ;ACCES NORMAL SI PAS IDENTIQUE
146  000125' 201044  000000  RNGMEM: MOVEI   TAC,(ADR)        ;DERNIERE ADRESSE
147  000126' 311040  000075•         CAML    TAC,SEGADR      ;ENGLOBEE PAR LE SEGMENT ?
148  000127' 311040  000070•         CAML    TAC,SEGZOR      ;... ?
149  000130' 263740  000000          POPJ    PIL,0           ;NON
150  000131' 563000  000123•         HRROS   SEGBLK          ;DRAPEAU POUR SIGNALER QUE LE BLOC DEVRA ETRE REECRIT
151  000132' 263740  000000          POPJ    PIL,0
152  000133' 376131  743100  MEMMES: ASCIZ   .?EOF EN ACCES MEMOIRE - INSTRUCTION.
     000134' 426344  040606
     000135' 416132  320232
     000136' 426331  744644
     000137' 425005  520222
     000140' 472472  451252
     000141' 416511  147634
     000142' 000000  000000
153
154                                 ;ROUTINE APPELEE AU MOMENT DE REPASSER LE CONTROLE A L'UTILISATEUR
155
156  000143' 200040  000077• CHKFRG: MOVE    TAC,SEGPAG      ;ADRESSE DU FRAGMENT LOCAL
157  000144' 316040  000062•         CAMN    TAC,PAGADR      ;DE LA PAGE EN COURS ?
158  000145' 331100  000131•         SKIPL   TAC1,SEGRLK     ;OUI - MERITE-T-IL RECOPIE ?
159  000146' 263740  000000          POPJ    PIL,0           ;NON
160  000147' 254000  000037'         JRST    BLTS01          ;OUI - METTRE LA PAGE A JOUR
161
```

```
                                SUBTTL  MODIFICATIONS D'ADRESSES

                                ;MISE EN RELATIF DE CE QUI EST POINTE PAR DAT

000150' 261247  000000  DATRL0: PUSH    DAT,(LOC)       ;RANGEMENT DU MOT
000151' 570045  000000  DATREL: HRRE    TAC,(DAT)       ;PARTIE ADRESSE
000152' 325040  000117*         JUMPGE  TAC,CPOPJ       ;INUTILE SI C'EST 0
000153' 271040  400000          ADDI    TAC,LOALOC      ;CAUSE 0 A GAUCHE
000154' 230040  000115*         IDIV    TAC,PAGLNG      ;CONVERSION
000155' 271100  400000          ADDI    TAC1,LOALOC     ;RAJOUTER LE SEUIL DE ZONE HAUTE
000156' 542105  000000          HRRM    TAC1,(DAT)      ;MISE EN PLACE DE LA QUANTITE RELATIVE
000157' 263740  000000          POPJ    PIL,0           ;ET RETOUR

                                ;MISE EN ABSOLU DE CE QUI EST POINTE PAR DAT SAUF SI 0

000160' 262247  000000  DATAB0: POP     DAT,(LOC)       ;DEPLACEMENT DU MOT
000161' 260740  000234'         PUSHJ   PIL,LOCREL      ;DE LOCAL EN RELATIF
000162' 550045  000000  DATABS: HRRZ    TAC,(DAT)       ;PARTIE ADRESSE
000163' 322040  000152*         JUMPE   TAC,CPOPJ       ;INUTILE SI 0
000164' 315040  000000*         CAMGE   TAC,PAG0LN      ;EN PAGE COURANTE ?
000165' 626040  400000          TRZN    TAC,LOALOC      ;... ?
000166' 263740  000000          POPJ    PIL,0           ;NON
000167' 270040  000144*         ADD     TAC,PAGADR      ;MISE EN ABSOLU COMPTE TENU DE LA PAGE COURANTE
000170' 542045  000000          HRRM    TAC,(DAT)       ;RANGEMENT DE LA PARTIE ADRESSE MISE A JOUR
000171' 263740  000000          POPJ    PIL,0           ;ET RETOUR

                                ;ROUTINE DE MISE EN ABSOLU DE ADR

000172' 550200  000016  UOADAB: HRRZ    ADR,UUO         ;MISE EN ABSOLU DE L'ADRESSE DES ARGUMENTS DE L'UUO
000173' 621000  000010  METABS: TLZ     DRP,NORPAG      ;SUPPOSE NON-MISE EN ABSOLU
000174' 606200  400000          TRNN    ADR,LOALOC      ;<LOALOC ?
000175' 263740  000000          POPJ    PIL,0           ;OUI
000176' 261740  000001          PUSH    PIL,TAC         ;POUR TRAVAIL
000177' 201044  400000          MOVEI   TAC,LOALOC(ADR) ;PARTIE DROITE SEULE - LOALOC
000200' 311040  000154*         CAML    TAC,PAGLNG      ;DANS LA PAGE 0 ?
000201' 254000  000000*         JRST    TPOPJ           ;NON - DEJA ABSOLU
000202' 620200  400000          TRZ     ADR,LOALOC      ;-LOALOC
000203' 270200  000167*         ADD     ADR,PAGADR      ;OUI - AJOUTER L'ADRESSE DE PAGE COURANTE
000204' 661000  000010          TLO     DRP,NORPAG      ;MISE EN ABSOLU (PB LIE AUX AMBIGUITES)
000205' 254000  000201*         JRST    TPOPJ           ;RESTAURER TAC ET RETOUR

                                ;ROUTINE DE MISE EN ABSOLU DES 2 MOITIES DE TAC

000206' 261740  000004  TTACAB: PUSH    PIL,ADR         ;POUR UTILISER METABS
000207' 554200  000001          HLRZ    ADR,TAC         ;PARTIE GAUCHE
000210' 260740  000217'         PUSHJ   PIL,LOCABS      ;...
000211' 504040  000004          HRL     TAC,ADR         ;C'EST FAIT
000212' 550200  000001          HRRZ    ADR,TAC         ;PARTIE DROITE
000213' 260740  000217'         PUSHJ   PIL,LOCABS      ;...
000214' 540040  000004          HRR     TAC,ADR         ;TERMINE
000215' 262740  000004          POP     PIL,ADR         ;RECUPERER L'ORIGINAL
000216' 263740  000000          POPJ    PIL,0           ;ET RETOUR
000217' 301200  400000  LOCABS: CAIL    ADR,LOALOC      ;EST-CE EN PAGE COURANTE ?
000220' 311200  000164*         CAML    ADR,PAG0LN      ;... ?
000221' 263740  000000          POPJ    PIL,0           ;NON
000222' 207000  000007          MOVSS   LOC             ;PARTIE ADRESSE DE PAGE A DROITE
000223' 271207  400000          ADDI    ADR,400000(LOC) ;- LE SEUIL DE CHARGEMENT + L'ADRESSE DE PAGE
000224' 207000  000007          MOVSS   LOC             ;RETOUR A L'ETAT INITIAL DE LOC
000225' 263740  000000          POPJ    PIL,0           ;RETOUR
                                ;MISE EN LOCAL DE CE QUI EST POINTE PAR DAT (DEJA EN RELATIF)

000226' 550045  000000  RELLOC: HRRZ    TAC,(DAT)       ;LA PARTIE DROITE SEULE
000227' 322040  000163*         JUMPE   TAC,CPOPJ       ;PAS RELATIF SI 0
000230' 275040  400000          SUBI    TAC,LOALOC      ;SEUIL
000231' 271040  002000          ADDI    TAC,PAGLOC      ;LA OU C'EST IMPLANTE
000232' 542045  000000          HRRM    TAC,(DAT)       ;RANGEMENT DE L'ADRESSE LOCALE
000233' 263740  000000          POPJ    PIL,0           ;ET RETOUR

                                ;MISE EN RELATIF DE L'ADRESSE LOCALE EN LA PAGE

000234' 550045  000000  LOCREL: HRRZ    TAC,(DAT)       ;LA PARTIE DROITE SEULE
000235' 322040  000227*         JUMPE   TAC,CPOPJ       ;PAS D'ADRESSE SI 0
000236' 275040  002000          SUBI    TAC,PAGLOC      ;LA OU C'EST IMPLANTE
000237' 271040  400000          ADDI    TAC,LOALOC      ;AJOUTER LE SEUIL
000240' 542045  000000          HRRM    TAC,(DAT)       ;RANGEMENT DE L'ADRESSE RELATIVE
000241' 263740  000000          POPJ    PIL,0           ;ET RETOUR
```

```
                                SUBTTL  ROUTINE D'EXTENSION DE LA MEMOIRE SUR LE DISQUE

000242' 200046  000052* CORUU0: MOVE    TAC,ACCUS(CHN)  ;ADRESSE DESIREE
000243' 322040  000305'         JUMPE   TAC,COREST      ;SIMPLE RENSEIGNEMENT

                                ;PARTIE UTILISEE PAR CE PROGRAMME POUR EXTENSION DE MEMOIRE

000244' 603040  400000  CORUU0: TLNE    TAC,400000      ;AUSSI POUR MI-SEG ?
000245' 254000  000000*         JRST    ILEGAL          ;OUI - PAS POSSIBLE
000246' 621040  777777          TLZ     TAC,-1          ;BITS INUTILES
000247' 261740  000073*         PUSH    PIL,USRRFL      ;SAUVER LA VALEUR INITIALE
000250' 317040  000247*         CAMG    TAC,USRREL      ;PLUS QU'ACTUELLEMENT ?
000251' 254000  000304'         JRST    CORES1          ;NON - OK
000252' 435040  001777          IORI    TAC,1777        ;ARRONDI AU K SUPERIEUR -1
000253' 552040  000250*         HRRZM   TAC,USRREL      ;MISE A JOUR COMME SI SUCCES
000254' 260740  000111'         PUSHJ   PIL,ARRSUP      ;ARRONDIR A LA PAGE SUPERIEURE
000255' 317057  000000          CAMG    TAC,(PIL)       ;EXTENSION ?
000256' 254000  000305'         JRST    COREST
000257' 336240  000000*         SKIPN   DAT,ABSMAX      ;LIMITE POSEE A-PRIORI ?
000260' 254000  000263'         JRST    CORUU2          ;NON...
000261' 311040  000005          CAML    TAC,DAT         ;OUI - RESPECTEE ?
000262' 254000  000315'         JRST    CORERR          ;NON...
000263' 275040  400000  CORUU2:SUBI     TAC,LOALOC      ;SOUSTRAIRE LE SEUIL DE CHARGEMENT
000264' 230040  000200*         IDIV    TAC,PAGLNG      ;CALCUL DU NO DE BLOC
000265' 220040  000000*         IMUL    TAC,UTILNG      ;...
000266' 231040  000200          IDIVI   TAC,BLKSIZ      ;...NO DE BLOC
000267' 550100  000000*         HRRZ    TAC1,MAXBLK     ;DERNIER EXISTANT
000270' 276040  000002          SUBM    TAC,TAC1        ;VOULU-ANCIEN=ANCIEN
000271' 323100  000304'         JUMPLE  TAC1,CORES1     ;SI CA VA DEJA...
000272' 542040  000267*         HRRM    TAC,MAXBLK      ;MISE A JOUR
000273' 075000  002000  EXPCOR: USETO   MEMORY,10=MAXK   ;UN NMB TRES GRAND POUR AVOIR LE SUIVANT
000274' 057000  000000*         OUT     MEMORY,CORIOW   ;ESSAYER DE SORTIR DES 0
000275' 367100  000273'         SOJG    TAC1,EXPCOR     ;CONTINUER...
000276' 260740  000335'         PUSHJ   PIL,STATUS
000277' 323100  000304'         JUMPLE  TAC1,CORES1     ;SI TOUT S'EST BIEN PASSE...
000300' 213000  000002          MOVNS   TAC1            ;CORRIGER MAXBLK - MANQUE DE PLACE
000301' 272100  000272*         ADDM    TAC1,MAXBLK
000302' 262740  000253*         POP     PIL,USRRFL      ;RESTITUER L'ANCIENNE VALEUR
000303' 254000  000306'         JRST    CORES2          ;NON-SKIP-RETURN
000304' 350000  000000* CORES1: AOS     PAGEPC          ;SKIP-RETURN
000305' 262757  000000  COREST: POP     PIL,(PIL)       ;CORRIGER LA PILE
000306' 200040  000302* CORES2: MOVE    TAC,USRREL      ;MAX ACTUEL
000307' 271040  000001          ADDI    TAC,1
000310' 242040  000012          LSH     TAC,+D10        ;NMB DE K
000311' 201100  000200          MOVEI   TAC1,MAXK       ;MAXI POSSIBLE
000312' 275101  000000          SUBI    TAC1,(TAC)      ;DANS TAC CE QUI RESTE DE K
000313' 202106  000242*         MOVEM   TAC1,ACCUS(CHN) ;RANGER
000314' 263740  000000          POPJ    PIL,0           ;ET RETOUR

000315' 262740  000306* CORERR: POP     PIL,USRREL      ;RESTITUER L'ANCIENNE VALEUR
000316' 201000  000455'         MOVEI   CORMES          ;ADRESSE DU MESSAGE
000317' 254000  000110*         JRST    ILEGAB          ;...

                                SUBTTL  ECHANGES PAR OPERATEURS PROGRAMMES (CANAL 0)
                                ;ROUTINE DE LECTURE D'UN BLOC DE MEMOIRE

000320' 201040  000000* MEMGET: MOVEI   TAC,MEMBLP       ;ADRESSE DE L'ORDRE POUR 1 BLOC
000321' 060000  000017  ANYGET: SETSTS  MEMORY,DUMP      ;S'ASSURER DE NON-FIN DE FILE
000322' 074002  000002          USETI   MEMORY,JOABLK+1(TAC1);PRECISER LE NUMERO DE BLOC DESIRE
000323' 056001  000000          IN      MEMORY,(TAC)    ;ORDRE D'ENTREE
000324' 263740  000000          POPJ    PIL,0           ;PAS D'ENNUIS
000325' 061000  740000          STATO   MEMORY,IOBAD    ;ERREUR DE LECTURE ?
000326' 254000  000000*         JRST    CPOPJ1          ;NON - CE DOIT ETRE END-OF-FILE
000327' 254000  000335'         JRST    STATUS          ;OUI - VOIR QUELLE ERREUR...

                                ;ROUTINE D'ECRITURE D'UN BLOC DE MEMOIRE

000330' 201040  000320* MEMSOR: MOVEI   TAC,MEMBLP       ;ADRESSE DE L'ORDRE POUR 1 BLOC
000331' 060000  000017  ANYSOR: SETSTS  MEMORY,DUMP      ;REMISE A JOUR DU MODE
000332' 075002  000002          USETO   MEMORY,JOABLK+1(TAC1);PRECISER SUR QUEL BLOC
000333' 057001  000000          OUT     MEMORY,(TAC)    ;ORDRE DE SORTIE
000334' 263740  000000          POPJ    PIL,0           ;SI PAS D'ENNUIS

                                ;ROUTINE DE VERIFICATION DE L'ETAT DE LA FILE

000335' 201000  000000  STATUS: MOVEI   0
000336' 062000  000001          STATUS  MEMORY,TAC      ;DANS TAC
000337' 602040  400000          TRNE    TAC,IOIMP       ;4 TESTS DE CAUSE D'ERREUR
000340' 201000  000351'         MOVEI   IMPMES
000341' 602040  200000          TRNE    TAC,IODEV
000342' 201000  000355'         MOVEI   DEVMES
000343' 602040  100000          TRNE    TAC,IODAT
000344' 201000  000360'         MOVEI   DATMES
000345' 602040  040000          TRNE    TAC,IOBKTL
000346' 201000  000363'         MOVEI   BKLMES
000347' 326000  000000*         JUMPN   TAPMES
000350' 263740  000000          POPJ    PIL,0           ;RETOUR SI PAS D'ERREUR

000351' 376125  751500  IMPMES: ASCIZ   .?E/S IMPOSSIBLE.
        000352' 446332  047646
        000353' 516230  246212
        000354' 000000  000000
000355' 376110  553222  DEVMES: ASCIZ   .?DEVICE ERROR.
        000356' 416124  042644
        000357' 512372  200000
000360' 376110  152202  DATMES: ASCIZ   .?DATA ERROR.
        000361' 202132  251236
        000362' 510000  000000
000363' 375070  442500  BKLMES: ASCIZ   .?#DE BLOC TROP GRAND.
        000364' 412311  741500
        000365' 522451  750100
        000366' 436450  147210
        000367' 000000  000000
```

```
MEMOIRE - GESTION DU FICHIER MEMOIRE : PAGES ET FRAGMENTS  09.08.78      MACRO.V43 10:49  2-SEP-78 PAGE 5
MEMOIR.MAC       GESTION DES PAGES

                                 SUBTTL  GESTION DES PAGES
                                 ;ROUTINE DE CHANGEMENT DE PAGE ADR CONTIENT L'ADRESSE DESIREE
                                 ;ADR,TAC,TAC1,TAC2 SONT PROTEGES - SEUL LOC EST DETRUIT

 000370' 200340  000464' ACCPAG: MOVE    LOC,[XWD TAC,SVTAC]:POUR SAUVER QQ ACCUMULATEURS
 000371' 251340  000000*         BLT     LOC,SVTAC2        ;...
 000372' 201104  000000          MOVEI   TAC1,(ADR)        ;RECOPIE
 000373' 311100  000203*         CAML    TAC1,PAGADR       ;EST CE DANS LA PAGE COURANTE ?
 000374' 311100  000056*         CAML    TAC1,PAGZDR       ;... ?
 000375' 334040  000143*         SKIPA   TAC,SEGPAG        ;NON
 000376' 254000  000433'         JRST    ACPAG1            ;OUI
 000377' 312040  000373*         CAME    TAC,PAGADR        ;SEGMENT COURANT DANS CETTE PAGE ?
 000400' 254000  000403'         JRST    .+3               ;NON
 000401' 315000  000145*         CAMGE   SEGBLK            ;OUI - SEGMENT MERITE RECOPIE ?
 000402' 260740  000040'         PUSHJ   PIL,BLTSEG        ;OUI - RECOPIE SUR LE DISQUE
 000403' 201044  000000          MOVEI   TAC,(ADR)         ;VERIFIER VALIDITE DE L'ADRESSE
 000404' 313040  000315*         CAMLE   TAC,USRRFL        ;VERIFIER LA VALIDITD DE L'ADRESSE
 000405' 254000  000102'         JRST    ILMREF            ;TROP GRANDE
 000406' 275040  400000          SUBI    TAC,LOALOC        ;POUR TEST SUIVANT!
 000407' 323040  000433'         JUMPLE  TAC,ACPAG1        ;ADRESSE < LOALOC !!
 000410' 230040  000264*         IDIV    TAC,PAGLNG        ;CALCUL DU NO DE BLOC DESIRE
 000411' 275104  000000          SUBI    TAC1,(ADR)        ;-ADRESSE DE TETE DE PAGE
 000412' 213040  000002          MOVNS   TAC,TAC1          ;+ADRESSE DE TETE DE PAGE
 000413' 250040  000377*         EXCH    TAC,PAGADR        ;RANGER ADRESSE DE NOUVELLE PAGE
 000414' 274100  000220*         SUB     TAC1,PAGBLN       ;LA PAGE 0 N'EXISTE PAS
 000415' 242100  777771          LSH     TAC1,BKSHFT       ;CONVERSION EN NO DE BLOC DE DISQUE
 000416' 250100  000000*         EXCH    TAC1,PAGBLK       ;PRENDRE L'ANCIEN NO
 000417' 602040  777777          TRNE    TAC,-1            ;ADRESSE DE PAGE EXISTAIT ?
 000420' 260740  000445'         PUSHJ   PIL,SORPAG        ;OUI - (A CAUSE DU DEPART)
 000421' 200100  000416*         MOVE    TAC1,PAGBLK       ;NO DE TETE DE LA PAGE DESIREE
 000422' 260740  000440'         PUSHJ   PIL,GETPAG        ;ALLER LA CHERCHER
 000423' 200040  000375*         MOVE    TAC,SEGPAG        ;ADRESSE DE PAGE DU SEGMENT COURANT
 000424' 312040  000413*         CAME    TAC,PAGADR        ;SEGMENT COURANT DANS CETTE PAGE ?
 000425' 254000  000430'         JRST    .+3               ;NON
 000426' 315000  000401*         CAMGE   SEGBLK            ;OUI - SEGMENT MERITE RECOPIE ?
 000427' 260740  000037'         PUSHJ   PIL,BLTSG1        ;OUI - RECOPIE DANS LA PAGE
 000430' 550040  000424*         HRRZ    TAC,PAGADR        ;MISE A JOUR DE PAGZDR
 000431' 270040  000410*         ADD     TAC,PAGLNG        ;ADRESSE DE LA PAGE SUIVANTE
 000432' 202040  000374*         MOVEM   TAC,PAGZDR        ;...
 000433' 201344  002000  ACPAG1: MOVEI   LOC,PAGLOC(ADR):ADRESSE DESIREE ABSOLUE -...
 000434' 274340  000430*         SUB     LOC,PAGADR        ;DONNE LA POSITION DANS LA PAGE
 000435' 204140  000464'         MOVS    TAC2,[XWD TAC,SVTAC]:POUR LE BLT
 000436' 251140  000003          BLT     TAC2,TAC2         ;RECUPERER LES ACCUMULATEURS
 000437' 263740  000000          POPJ    PIL,0             ;ET RETOUR

 000440' 201040  000000* GETPAG: MOVEI   TAC,PAGBLP        ;ADRESSE DE L'ORDRE DUMP
 000441' 260740  000321'         PUSHJ   PIL,ANYGET        ;EXECUTION DE L'ENTREE
 000442' 263740  000000          POPJ    PIL,0             ;OK
 000443' 201000  000447'         MOVEI   GPGMES            ;ERREUR END-OF-FILE
 000444' 254200  000317*         HALT    ILEGA0            ;IMPRIMER LE MESSAGE
                                 ;SORTIE DE LA PAGE

 000445' 201040  000440* SORPAG: MOVEI   TAC,PAGBLP        ;ADRESSE DE L'ORDRE DUMP
 000446' 254000  000331'         JRST    ANYSOR            ;EXECUTION DE LA SORTIE

 000447' 376131  743100  GPGMES: ASCIZ   ;?EOF EN LECTURE PAGE - UUO.
         000450' 426344  046212
         000451' 416512  551212
         000452' 202410  143612
         000453' 201324  052652
         000454' 474000  000000
 000455' 376311  146622  CORMES: ASCIZ   ;?LIMITE DE MEMOIRE ATTEINTE - UUO.
         000456' 522124  042212
         000457' 202330  546636
         000460' 446450  520202
         000461' 522510  544634
         000462' 522124  026500
         000463' 526531  700000

 000464'                 LIT
 000464' 000001  000000*
 000465'                 VAR
                         END

NO ERRORS DETECTED

PROGRAM BREAK IS 000465

MEMOIRE - GESTION DU FICHIER MEMOIRE : PAGES ET FRAGMENTS  09.08.78      MACRO.V43 10:49  2-SEP-78 PAGE 6
MEMOIR.MAC       SYMBOL TABLE

ABSMAX          000257' EXT     ACCMEM          000002' INT     ACCMM1          000000' INT     ACCPAG          000370' INT
ACCUS           000313' EXT     ACMMAB          000001' INT     ACPAG1          000433'         ACU             000010  SPD
ADR             000004  SPD     ALAPRB          000003  SPD     ANYGET          000321' INT     ANYSOR          000331' INT
ARRSUP          000111' INT     ASCIIL          000001  SPD     BKLMES          000363'         BKSHFT  777777  777771  SPD
BLKSIZ          000200  SPD     BLTSEG          000040'         BLTSG1          000037'         CHKFRG          000143' INT
CHKMEM          000050' INT     CHN             000006  SPD     CKMEM1          000066'         CKMEM2          000100'
CORE            000011  SPD     CORERR          000315'         CORES1          000304'         CORES2          000306'
COREST          000305'         CORIOW          000274' EXT     CORMES          000455'         CORUUO          000244' INT
CORUU2          000263'         CORUUO          000242' INT     CPOPJ           000235' EXT     CPOPJ1          000326' EXT
DAT             000005  SPD     DATAB0          000160' INT     DATABS          000162' INT     DATMES          000368'
DATREL          000151' INT     DATRL0          000150' INT     DDTOUT          000003  SPD     DEVCHR          000001  SPD
DEVMES          000355'         DRP             000000  SPD     DUMP            000017  SPD     EXIT            000012  SPD
EXPCOR          000273'         GETPAG          000440'         GPGMES          000447'         HGHOFF          000002  SPD
HGHON           000003  SPD     HIGHSW  777777  777775  SPD     HORPAG          000004  SPD     IOX             000011  SPD
ILEGA0          000444' EXT     ILEGAL          000245' EXT     ILM             020000  SPD     ILMENB          000016  SPD
ILMREF          000102'         IMMB            000001  SPD     IMPMES          000351'         INB             000016  SPD
IOBAD           740000  SPD     IOBKTL          040000  SPD     IODAT           100000  SPD     IODEV           200000  SPD
IOIMP           400000  SPD     JOABLK          000001  SPD     JOBCNI          000103' EXT     JOBPAG          000140  SPD
LOALOC          400000  SPD     LOC             000007  SPD     LOCABS          000217'         LOCREL          000234'
MAXBLK          000301' EXT     MAXK            000200  SPD     MEMBLP          000338' EXT     MEMGET          000320' INT
MEMMES          000133'         MEMORY          000000  SPD     MEMSOR          000330' INT     METABS          000173' INT
NADRB           000002  SPD     NORPAG          000010  SPD     OPTESS  777777  777777  SPD     PAGBLN          000414' EXT
PAGADR          000434' EXT     PAGBLK          000421' EXT     PAGBLP          000449' EXT     PAGEPC          000304' EXT
PAGLNG          000431' EXT     PAGLOC          002000  SPD     PAGZDR          000432' EXT     PEEK            000033  SPD
PIL             000017  SPD     RELLOC          000226' INT     RNGMEM          000125' INT     RNGMN1          000120' INT
RNGMN0          000122' INT     RNGMN1          000121' INT     SD              000015  SPD     SEGADR          000126' EXT
SEGBLK          000426' EXT     SEGLOC          000076' EXT     SEGPAG          000423' EXT     SEGSIZ          000200  SPD
SEGZDR          000127' EXT     SETDDT          000002  SPD     SGSHFT  777777  777771  SPD     SLEEP           000031  SPD
SORPAG          000445'         SORSEG          000032' INT     SORSG0          000034'         STATUS          000335' INT
SVTAC           000464' EXT     SVTAC2          000371' EXT     TAC             000001  SPD     TAC1            000002  SPD
TAC2            000003  SPD     TAPMES          000347' EXT     TPOPJ           000205' EXT     TTACAB          000206' INT
TTY             000000  SPD     UOADAB          000172' INT     USRENB          000104' EXT     USRREL          000404' EXT
USRTRP          000106' EXT     UTILNG          000265' EXT     UUO             000016  SPD     UUOOFF          000001  SPD
UUOON           000000  SPD     UUOSW   777777  777775  SPD

2K CORE USED
```

ABSMAX	26#	256												
ACCMEM	20	50#	145											
ACCMM1	20	47#												
ACCPAG	22	334#												
ACCUS	26#	94	240	264										
ACMMAB	22	49#												
ACPAG1	340	349	369#											
ACU	5#													
ADR	5#	47	63	92	141	142	146	190	192	195	198	199	205	206
	208	209	211	212	214	215	216	336	345	351	369			
ALAPRB	5#													
ANYGET	21	295#	376											
ANYSOR	21	306#	383											
ARRSUP	22	132#	253											
ASCIIL	5#													
BKLMES	322	329#												
BKSHFT	5#	34	355											
BLKSIZ	5#	32	263											
BLTSEG	83#	344												
BLTSG1	82#	168	365											
CHKFRG	22	156#												
CHKMEM	21	50	92#	143										
CHN	5#	240	284											
CKMEM1	99	107#												
CKMEM2	96	106	117#											
CORE	5#													
CORERR	259	287#												
CORES1	250	266	272	277#										
CORES2	276	279#												
CORES T	241	255	278#											
CORIOW	28#	269												
CORMES	288	306#												
CORUU0	21	245#												
CORUU2	257	260#												
CORUUO	21	240#												
CPOPJ	27#	138	168	180	224	233								
CPOPJ1	27#	300												
DAT	5#	166	167	172	177	179	185	223	227	232	236	256	258	
DATAB0	21	177#												
DATABS	21	179#												
DATMES	320	328#												
DATREL	22	167#												
DATRL0	22	166#												
DDTOUT	5#													
DEVCHR	5#													
DEVMES	318	327#												
DHP	5#	103	104	107	191	200								
DUMP	5#	295	306											
EXIT	5#													
EXPCOR	268#	270												
GETPAG	360	375#												
GPGMES	378	385#												
HGHOFF	5#													
HGHON	5#													
HIGHSW	5#													
HORPAG	5#	13#	104	107										
IDX	5#													
ILEGA0	27#	73	128	289	379									
ILEGAL	27#	246												
ILM	5#													
ILMENB	5#													
ILMREF	113	122#	347											
IMMB	5#	5												
IMPMES	316	326#												
INS	5#													
IOBAD	18#	299												
IOBKTL	17#	18	321											
IODAT	16#	18	319											
IODEV	15#	18	317											
IOIMP	14#	18	315											
JDABLK	5#	296	307											
JOBCNI	27#	123												
JOBPAG	5#													
LOALOC	5#	33#	57	60	62	111	132	137	169	171	182	192	195	198
	214	225	235	260	348									
LOC	5#	53	54	55	56	57	62	63	64	65	66	92	93	94
	95	97	98	100	101	102	108	109	111	112	114	115	116	122
	123	124	125	166	177	217	218	219	334	335	369	370		
LOCABS	207	210	214#											
LOCREL	178	232#												
LSTCB	5													
MAXBLK	26#	264	267	274										
MAXK	30#	268	282											
MEMBLP	25#	294	305											
MEMGET	20	70	294#											
MEMMES	72	127	152#											
MEMORY	5#	268	269	295	296	297	299	306	307	308	314			
MEMSOR	20	81	305#											
METABS	22	49	191#											
NADRB	5#	5												
NORPAG	5#	103	191	200										
OPTESS	5													
PAGBLN	28#	181	215	354										
PAGADR	25#	80	97	100	102	157	184	199	337	341	353	362	366	370
PAGBLK	26#	356	359											
PAGBLP	26#	375	382											
PAGEPC	28#	277												
PAGLNG	25#	58	59	67	133	136	170	196	261	350	367			
PAGLOC	5#	87	89	95	101	226	234	369						
PAGZOR	25#	98	338	368										
PEEK	5#													
PIL	5#	49	50	52	70	71	90	110	117	118	143	149	151	159
	173	178	183	186	193	194	205	207	210	212	213	216	220	228
	237	248	253	254	271	275	278	285	287	298	300	324	344	350
	360	365	373	376	377									
RELLOC	23	223#												
RNGMEM	20	146#												
RNGMM1	21	141#												
RNGMN0	22	143#												
RNGMN1	20	142#												
SD	5#													
SEGADR	25#	54	85	108	114	147								
SEGBLK	25#	51	69	77	83	144	150	158	343	364				
SEGLOC	26#	65	88	115										
SEGPAG	28#	61	78	79	84	116	156	339	361					
SEGSIZ	32#	53	55	57	62	63	89							
SEGZOR	25#	56	82	109	148									

SETDDT	5#													
SGSHFT	34#	68												
SLEEP	5#													
SORPAG	358	382#												
SORSEG	20	77#												
SORSGØ	52	79#												
STATUS	20	271	301	313#										
SVTAC	26#	334	371											
SVTAC2	26#	335												
TAC	5#	5	57	58	59	60	61	66	79	80	85	86	89	132
	133	134	135	136	137	138	146	147	148	156	157	167	168	169
	170	179	180	181	182	184	185	194	195	196	206	208	209	211
	223	224	225	226	227	232	233	234	235	236	240	241	245	247
	249	251	252	254	258	260	261	262	263	265	267	279	280	281
	283	294	297	305	308	314	315	317	319	321	334	339	341	345
	346	348	349	350	352	353	357	361	362	366	367	368	371	375
	382													
TAC1	5#	51	62	67	68	69	86	87	88	89	134	158	171	172
	264	265	266	270	272	273	274	282	283	284	296	307	336	337
	338	351	352	354	355	356	359							
TAC2	5#	371	372											
TAPMES	27#	323												
TPOPJ	27#	197	201											
TTACAB	23	205#												
TTY	5#													
UDADAB	23	190#												
USRENB	28#	124												
USRREL	28#	112	248	249	252	275	279	287	346					
USRTRP	27#	126												
UTILNG	25#	262												
UUO	5#	5	190											
UUOOFF	5#													
UUOON	5#													
UUOSW	5#	5												

```
ENTSOR  OPERATEURS PROGRAMMES D'ENTREE-SORTIE 09.08.70  MACRO.V43 23:37  2-SEP-70 PAGE 1
CB    .MAC      DECLARATIONS

                        ;CETTE FILE A ETE ASSEMBLEE AVEC LA FILE DES PARAMETRES COMMUNS
                        ;AUX SOUS-PROGRAMMES DE PAGINATION  CB.MAC

                        XLIST
                        LIST
                        TITLE   ENTSOR  OPERATEURS PROGRAMMES D'ENTREE-SORTIE 09.08.70
                        SUBTTL  DECLARATIONS

                        ;ASSEMBLAGE AVEC LA FILE DES PARAMETRES CB.MAC

                        .DRAPEAUX

                400000  SKIPSW==          400000
                200000  BUFSW==           200000  ;LA LONGUEUR D'UN TAMPON A DEJA ETE DEMANDEE AU MONITAUR
                100000  IO==              100000  ;SORTIE ET NON ENTREE

                        .QUANTITE POUR L'UUO TTCALL

                000003  OUTSTR==          3       ;SORTIE D'UNE CHAINE ASCIZ

                        .QUANTITES DEPENDANT DU MONITEUR 10-50

                002000  ICLOSB==          2000
                001000  OCLOSB==          1000    ;FILE NON FERMEE EN SORTIE

                        INTERN  UOPEN   ,UINIT  ,UFILE  ,UIFILE ,UOFILE ,CLSUUO ,RLSUUO ,CNVCAN
                        INTERN  UINBF   ,UOUTBF ,TTYUUO ,IORELE ,STSSET ,STSGET ,TIN    ,TOUT

                        EXTERN  TAPMES  ,ACCMEM ,ACCMM1 ,ACCPAG ,RNGMN1 ,RNGMEM ,CORUUO ,XCTUUO
                        EXTERN  PTACAC  ,PAGMTR ,CPOPJ  ,PAGBLN ,PURUUO ,PUUOAC ,SEGBLK ,SORSEG
                        EXTERN  PURARG  ,PUROPE ,UMMTAB ,PAGEPC ,PURMMI ,PURMMO ,PURMMA ,PURMMZ
                        EXTERN  ACCUS   ,ALLCAN ,PAGLNG ,UTILNG ,PAGADR ,PAGZDR ,MEMBLK ,DATABB
                        EXTERN  JOBFF   ,JOBSA  ,ILEGAB ,DATABS ,DATREL ,DATRLB ,METABS ,RELLOC
                        EXTERN  USRSA   ,USRFF  ,TTACAB ,UOADAB

                        INTERN  JOBMCU  ,JOBJDA
                000072  JOBMCU==         72
                000075  JOBJDA==         75

                400000  LOALOC==400000  ;DEBUT DES ADRESSES DE L'UTILISATEUR

000000'                 HISEG
                        HLON
```

```
                                SUBTTL  INIT - OPEN - RELEASE

                        .DISPATCH ICI APRES RECONNAISSANCE DE L'UUO INIT

000000' 550200  000000* UINIT:  HRRZ    ADR,PAGEPC      ;ADRESSE DE L'INSTRUCTION EN MEMOIRE
000001' 550040  000016          HRRZ    TAC,UUO         ;PREMIER ARGUMENT = MODE OU STATUS
000002' 260740  000130'         PUSHJ   PIL,RNGAR1      ;RANGER LES ARGUMENTS
000003' 641700  011000          TLC     UUO,011000      ;TRANSFORMER INIT EN OPEN
000004' 350000  000000*         AOS     PAGEPC          ;SAUTER 2EME ARGUMENT
000005' 350000  000004*         AOS     PAGEPC          ;SAUTER LE 3EME ARGUMENT
000006' 254000  000011'         JRST    UOPEN0          ;ET CONTINUER COMME L'UUO OPEN

                        .DISPATCH ICI APRES RECONNAISSANCE DE L'UUO OPEN

000007' 260740  000000* UOPEN:  PUSHJ   PIL,UOADAB       ;ADRESSE DES ARGUMENTS
000010' 260740  000126'         PUSHJ   PIL,RNGARG      ;RANGER EN ZONE PURE LES ARGUMENTS
000011' 201040  000017  UOPEN0: MOVEI   TAC,17
000012' 404040  000000*         AND     TAC,PURARG      ;MODE SEUL
000013' 301040  000015          CAIL    TAC,SD          ;EST-CE DUMP ?
000014' 402006  000000*         SETZM   UMMTAB(CHN)     ;OUI - PAS D'ADRESSE DE TETES DE TAMPONS
000015' 541700  000012*         HRRI    UUO,PURARG      ;ADRESSE DES ARGUMENTS D'OPEN
000016' 256000  000016          XCT     UUO             ;FAIRE L'UUO
000017' 334000  000000          SKIPA                   ;ERREUR
000020' 350000  000005*         AOS     PAGEPC          ;RETOUR CORRECT - SAUT D' 1 INSTRUCTION
000021' 254000  000235'         JRST    RNGMMD          ;RANGER LES TETES DE TAMPONS

                        ;ROUTINE POUR FAIRE UN RELEASE D'UN CANAL

000022' 260740  000153' IORELE: PUSHJ   PIL,IOPAG1      ;MANIPULER LES TAMPONS EVENTUELS
000023' 205040  071000          MOVSI   TAC,(RELEAS)    ;...
000024' 137300  000000*         DPB     CHN,PTACAC      ;NO DE CANAL EN BONNE PLACE
000025' 256000  000001          XCT     TAC             ;EXECUTER L'UUO
000026' 402006  000014*         SETZM   UMMTAB(CHN)     ;CANAL LIBERE
000027' 263740  000000          POPJ    PIL,0           ;EXIT
```

```
ENTSOR  OPERATEURS PROGRAMMES D'ENTREE-SORTIE 09:08.78  MACRO,V43 23:37  2-SEP-78 PAGE 2
ENTSOR.MAC      INBUF - OUTBUF - CONSTRUCTION DES SERIES DE TAMPONS

                                        SUBTTL  INBUF - OUTBUF - CONSTRUCTION DES SERIES DE TAMPONS

                                ;TRAITEMENT DES UUO'S INBUF ET OUTBUF

 000030' 625000  100000  UINBF:  TLZA    DRP,IO          ;OPERATION D'ENTREE
 000031' 661000  100000  UOUTBF: TLO     DRP,IO          ;OPERATION DE SORTIE
 000032' 201040  000001  IOBUF:  MOVEI   TAC,1           ;DORMIR UN PEU
 000033' 047040  000031          CALLI   TAC,SLEEP
 000034' 260740  000600'         PUSHJ   PIL,CNVCAN      ;CHANGER LE NO DE CANAL
 000035' 202700  000000* IOBF0C: MOVEM   UUO,PURUUO      ;SAUVER LA DEMANDE INITIALE
 000036' 332046  000026*         SKIPE   TAC,UHHTAB(CHN) ;DES TAMPONS ?
 000037' 260740  000201'         PUSHJ   PIL,PRDHHD      ;OUI - ALLER LES CHERCHER
 000040' 550200  000000* IOBF0B: HRRZ    ADR,USRFF       ;DERNIERE PAGE OU...
 000041' 260740  000000*         PUSHJ   PIL,METARS      ;MISE EN ABSOLU DANS CETTE PAGE AU BESOIN
 000042' 542200  000040*         HRRM    ADR,USRFF       ;RANGEMENT EN ABSOLU
 000043' 260740  000000* IOBF0A: PUSHJ   PIL,ACCPAG       ;L'ENTRER
 000044' 506340  000000*         HRLM    LOC,JOBSA       ;JOBFF LOCAL AVANT CREATION DES TAMPONS
 000045' 552340  000000*         HRRZM   LOC,JOBFF       ;JOBFF LOCAL
 000046' 260740  000114' IOBUF0: PUSHJ   PIL,BUFLNG      ;CHERCHER LA TAILLE D'UN TAMPON
 000047' 405040  007777          ANDI    TAC,7777        ;MASQUER LA TAILLE D'UN TAMPON
 000050' 334140  000001          SKIPA   TAC2,TAC        ;SAUVER LA TAILLE
 000051' 541716  777777  IOBUF1: HRRI    UUO,-1(UUO)     ;DECREMENTER LE NMB DE TAMPONS
 000052' 606700  777777          TRNN    UUO,-1          ;PAS 0 TAMPON ?
 000053' 254000  000121'         JRST    BUFERR          ;SI ...
 000054' 200040  000003          MOVE    TAC,TAC2        ;TAILLE
 000055' 221056  000000          IMULI   TAC,(UUO)       ;*NMB DE TAMPONS
 000056' 313040  000000*         CAMLE   TAC,UTILNG      ; < 1PAGE
 000057' 254000  000051'         JRST    IOBUF1          ;NON
 000060' 554100  000000*         HLRZ    TAC1,USRSA      ;USRFF INITIAL
 000061' 270040  000042*         ADD     TAC,USRFF       ;AJOUTER LA VALEUR ABSOLUE
 000062' 305202  000000          CAIGE   ADR,(TAC1)      ;JOBFF EN COURS DE PROGRAMME ?
 000063' 254000  000077'         JRST    IOBUF2          ;OUI...
 000064' 200100  000056*         MOVE    TAC1,UTILNG     ;LONGUEUR DE LA PAGE
 000065' 270100  000000*         ADD     TAC1,PAGADR     ;PLUS HAUTE ADRESSE POUR CETTE PAGE
 000066' 305042  000000          CAIGE   TAC,(TAC1)      ;ASSEZ POUR TENIR DANS LA PAGE ?
 000067' 254000  000103'         JRST    IOBUF3          ;OUI...
 000070' 261746  000000*         PUSH    PIL,ACCUS(CHN)  ;SAUVER AVANT CORUUO
 000071' 202206  000070*         MOVEM   ADR,ACCUS(CHN)  ;SIMULER
 000072' 260740  000000*         PUSHJ   PIL,CORUUO      ;INCREMENTER LA MEMOIRE
 000073' 262746  000071*         POP     PIL,ACCUS(CHN)  ;RESTAURER L'AC UTILISATEUR
 000074' 550200  000065*         HRRZ    ADR,PAGADR      ;L'ANCIENNE
 000075' 270200  000000*         ADD     ADR,PAGLNG      ;POUR LA SUIVANTE
 000076' 254000  000043'         JRST    IOBF0A          ;RECOMMENCER
 000077' 200100  000064* IOBUF2: MOVE    TAC1,UTILNG     ;LONGUEUR DE LA PAGE
 000100' 270100  000074*         ADD     TAC1,PAGADR     ;PLUS HAUTE ADRESSE POUR CETTE PAGE
 000101' 301042  000000          CAIL    TAC,(TAC1)      ;EST CE QUE CA TIENT DANS LA PAGE ?
 000102' 254000  000051'         JRST    IOBUF1          ;NON - REDUIRE LE NOMBRE DE TAMPONS
 000103' 202040  000061* IOBUF3: MOVEM   TAC,USRFF        ;NOUVELLE VALEUR (RELATIVE)
 000104' 256000  000016          XCT     UUO             ;EXECUTER L'UUO (PAS BESOIN DES AC'S)
 000105' 554040  000044*         HLRZ    TAC,JOBSA       ;INITIAL AVANT CREATION DE CES TAMPONS
 000106' 271040  000001          ADDI    TAC,1           ;+1  POUR POINTER SUR LE SECOND MOT DU
 000107' 603000  100000          TLNE    DRP,IO          ;PREMIER TAMPON - OUTBUF ?
 000110' 542040  000000*         HRRM    TAC,PURHHO      ;OUI - VALEUR LOCALE
 000111' 607000  100000          TLNN    DRP,IO          ;INBUF ?
 000112' 542040  000000*         HRRM    TAC,PURHHI      ;OUI - VALEUR LOCALE
 000113' 254000  000235'         JRST    RNGHHD          ;RANGER LES TETES DE TAMPONS

 000114' 550046  000075  BUFLNG: HRRZ    TAC,JOBJDA(CHN) ;ADRESSE DU DDB
 000115' 271040  000001          ADDI    TAC,DEVCHR      ;DDB+???
 000116' 047040  000033          CALLI   TAC,PEEK        ;LIRE DANS LE MONITEUR
 000117' 271040  000002          ADDI    TAC,2           ;+2 D'EN-TETE
 000120' 263740  000000          POPJ    PIL,0           ;ET RETOUR

 000121' 370700  000035* BUFERR: SOS     UUO,PURUUO      ;VALEUR INITIALE -1
 000122' 606700  777777          TRNN    UUO,-1  ;ENCORE ASSEZ ?
                                 JRST    [MOVEI  CASCIZ  ,ENNUIS 0 TAMPON.]
 000123' 254000  000665'                 JRST    [LEGA0]
 000124' 272140  000103*         ADDM    TAC2,USRFF      ;COMME SI UN AVAIT ETE ACCORDE
 000125' 254000  000040'         JRST    IOBF0B          ;CE DEVRAIT ALLER CETTE FOIS

                                        SUBTTL  ROUTINES COMMUNES POUR DEPLACER LES TETES DE TAMPONS

                                ;SOUS ROUTINE DE RANGEMENT DES ARGUMENTS DE L'UUO OPEN

 000126' 260740  000000* RNGARG: PUSHJ   PIL,ACCMEM      ;ACCES MEMOIRE
 000127' 200047  000000          MOVE    TAC,(LOC)       ;PRENDRE LE PREMIER ARGUMENT

                                ;SOUS ROUTINE DE RANGEMENT DES ARGUMENTS DES UUOS OPEN ET INIT

 000130' 202040  000015* RNGAR1: MOVEM   TAC,PURARG      ;LE RANGER EN PURARG EN ZONE PURE
 000131' 260740  000600'         PUSHJ   PIL,CNVCAN      ;CHANGER LE NUMERO DE CANAL
 000132' 260740  000000*         PUSHJ   PIL,ACCMM1      ;ACCES A LA POSITION DE MEMOIRE SUIVANTE
 000133' 200047  000000          MOVE    TAC,(LOC)       ;NOM DU PERIPHERIQUE
 000134' 202040  000001*         MOVEM   TAC,PURARG+1    ;SECOND ARGUMENT DE L'UUO
 000135' 260740  000132*         PUSHJ   PIL,ACCMM1      ;ACCES A LA POSITION DE MEMOIRE SUIVANTE
 000136' 200047  000000          MOVE    TAC,(LOC)       ;ADRESSES DES TETES DE TAMPONS
 000137' 260740  000000*         PUSHJ   PIL,TTACAB      ;MISE EN ABSOLU
 000140' 250046  000036*         EXCH    TAC,UHHTAB(CHN) ;RANGEMENT DES ADRESSES DE TETES DE TAMPONS
 000141' 332000  000001          SKIPE   TAC             ;DEJA UTILISE ?
 000142' 260740  000174'         PUSHJ   PIL,IOPAG2      ;OUI - FAIRE LE RELEASE PREALABLE
 000143' 200046  000140*         MOVE    TAC,UHHTAB(CHN) ;PREPARER LES PURES TETES DE TAMPON
 000144' 201100  000000          MOVEI   TAC1,0          ;SUPPOSER DUMP A-PRIORI
 000145' 603040  777777          TLNE    TAC,-1          ;TAMPONS DE SORTIE ?
 000146' 505100  000110*         HRLI    TAC1,PURHHO     ;OUI - PRECISER PUR TAMPON DE SORTIE
 000147' 602040  777777          TRNE    TAC,-1          ;TAMPONS D'ENTREE ?
 000150' 541100  000112*         HRRI    TAC1,PURHHI     ;OUI - PRECISER PURE TETE DE TAMPON D'ENTREE
 000151' 202100  000002*         MOVEM   TAC1,PURARG+2   ;TROISIEME ARGUMENT DE L'UUO
 000152' 254000  000201'         JRST    PRDHHD          ;PLACER EN MEMOIRE LES TETES DE TAMPON
```

```
                                ;SOUS ROUTINE DE MISE EN PLACE DE LA BONNE PAGE POUR UN RELEASE EVENTUEL

000153' 336046  000143* IOPAG1: SKIPN   TAC,UHHTAB(CHN) ;DES TAMPONS ?
000154' 263740  000000          POPJ    PIL,0           ;NON...
000155' 201100  000001  IOPAG3: MOVEI   TAC1,1          ;POUR RESCHEDULER...
000156' 047100  000031          CALLI   TAC1,SLEEP      ;...
000157' 332006  000075          SKIPE   JOBJDA(CHN)     ;CE CANAL EST-IL DEJA INITIALISE ?
000160' 313300  000072          CAMLE   CHN,JOBHCU      ;D'APRES LES RENSEIGNEMENTS DU MONITEUR
000161' 263740  000000          POPJ    PIL,0           ;NON...RETOUR IMMEDIAT
000162' 261740  000004          PUSH    PIL,ADR         ;SAUVER DES CHOSES...
000163' 260740  000201'         PUSHJ   PIL,PRDHWD      ;CHARGER LES TETES DE TAMPON ADEQUATES
000164' 260740  000254'         PUSHJ   PIL,IOPAGE      ;PRENDRE LA BONNE PAGE POUR LES TAMPONS
000165' 262740  000004          POP     PIL,ADR         ;RESTAURER ADR
000166' 263740  000000          POPJ    PIL,0           ;ET RETOUR

000167' 260740  000600' RLSUUO: PUSHJ   PIL,CNVCAN       ;POUR COMMENCER
000170' 201100  000000          MOVEI   TAC1,0          ;POUR NETTOYER L'ALLOCATION DE CANAL
000171' 137100  000001          DPB     TAC1,TAC        ;(SELON CNVCAN)
000172' 250106  000153*         EXCH    TAC1,UHHTAB(CHN);ET LA TABLE DES TETES DE TAMPONS
000173' 332040  000002          SKIPE   TAC,TAC1        ;SAUT SI PAS DE TAMPONS PRECISES
000174' 260740  000155' IOPAG2: PUSHJ   PIL,IOPAG3       ;ACCEDER A LA BONNE PAGE
000175' 205040  071000          MOVSI   TAC,(RELEAS)    ;INSTRUCTION A EXECUTER
000176' 137300  000024*         DPB     CHN,PTACAC      ;MISE EN PLACE DU NO DE CANAL
000177' 256000  000001          XCT     TAC             ;EXECUTER LE RELEASE
000200' 263740  000000          POPJ    PIL,0           ;ET RETOUR A L'APPEL

                                ;SOUS ROUTINE DE MISE EN PLACE DES TETES DE TAMPONS

000201' 403140  000002* PRDHWD: SETZB   TAC2,PURHWA      ;MISE A ZERO A-PRIORI
000202' 200100  000667'         MOVE    TAC1,[XWD PURHWA,PURHWA+1]
000203' 251100  000000*         BLT     TAC1,PURHWZ     ;DE LA ZONE PURE
000204' 550200  000001          HRRZ    ADR,TAC         ;TETE DE TAMPON D'ENTREE
000205' 322200  000210'         JUMPE   ADR,PRDHWO      ;INUTILE SI 0
000206' 551240  777777*         HRRZI   DAT,PURHWI-1    ;ADRESSE DE RANGEMENT
000207' 260740  000214'         PUSHJ   PIL,MOVHWD      ;DEPLACEMENT DES 3 ARGUMENTS
000210' 207000  000003  PRDHWO: MOVSS   TAC2            ;POUR L'AUTRE MOITIE
000211' 554200  000001          HLRZ    ADR,TAC         ;TETE DE TAMPON DE SORTIE
000212' 322200  000000*         JUMPE   ADR,CPOPJ       ;INUTILE SI 0
000213' 551240  777777*         HRRZI   DAT,PURHWO-1    ;ADRESSE DE SORTIE
000214' 261740  000001  MOVHWD: PUSH    PIL,TAC         ;SAUVER LES TETES DE TAMPONS
000215' 261740  000003          PUSH    PIL,TAC2        ;TETE DE PAGE DE CES TAMPONS
000216' 260740  000126*         PUSHJ   PIL,ACCMFM      ;ACCES MEMOIRE
000217' 260740  000000*         PUSHJ   PIL,DATRL0      ;RANGEMENT DU PREMIER TERME MIS EN RELATIF
000220' 220040  000075*         IMUL    TAC,PAGLNG      ;ADRESSE ABSOLUE DE TETE DE PAGE OU 0
000221' 322040  000223'         JUMPE   TAC,.+2 ;SAUF SI 0
000222' 271040  400000          ADDI    TAC,LOALOC      ;AJOUTER LE SEUIL DE CHARGEMENT
000223' 506057  000000          HRLM    TAC,(PIL)       ;POUR IOPAGE
000224' 260740  000000*         PUSHJ   PIL,RELLOC      ;DE RELATIF EN LOCAL
000225' 260740  000135*         PUSHJ   PIL,ACCMM1      ;ACCES AU SUIVANT
000226' 260740  000217*         PUSHJ   PIL,DATRL0      ;RANGEMENT DU DEUXIEME TERME MIS EN RELATIF
000227' 260740  000224*         PUSHJ   PIL,RELLOC      ;DE RELATIF EN LOCAL
000230' 260740  000225*         PUSHJ   PIL,ACCMM1      ;ACCES AU SUIVANT
000231' 261247  000000          PUSH    DAT,(LOC)       ;RANGEMENT DU TROISIEME TERME
000232' 262740  000003          POP     PIL,TAC2        ;RESTAURER TAC2
000233' 262740  000001  TPOPJ:: POP     PIL,TAC         ;RESTAURER TAC
000234' 263740  000000          POPJ    PIL,0           ;ET RETOUR

                                ;SOUS ROUTINE DE RANGEMENT DES TETES DE TAMPONS

000235' 550206  000172* RNGHWD: HRRZ    ADR,UHHTAB(CHN) ;ADRESSE ABSOLUE DE LA TETE DE TAMPON
000236' 322200  000241'         JUMPE   ADR,RNGHWO      ;D'ENTREE - INUTILE SI 0
000237' 561240  000002*         HRROI   DAT,PURHWI+2    ;STOCKAGE A L'ENVERS PAR POP
000240' 260740  000244'         PUSHJ   PIL,MVMHWD      ;RANGEMENT EN MEMOIRE
000241' 554206  000235* RNGHWO: HLRZ    ADR,UHHTAB(CHN) ;ADRESSE ABSOLUE DE LA TETE DE TAMPON
000242' 322200  000212*         JUMPE   ADR,CPOPJ       ;DE SORTIE - INUTILE SI 0
000243' 561240  000002*         HRROI   DAT,PURHWO+2    ;...
000244' 271200  000002  MVMHWD: ADDI    ADR,2           ;POUR PROCEDER EN REMONTANT
000245' 260740  000216*         PUSHJ   PIL,ACCMFM      ;BLOC CONCERNE
000246' 260740  000000*         PUSHJ   PIL,DATAR0      ;RANGEMENT ET MISE EN ABSOLU DU PRECEDENT
000247' 260740  000000*         PUSHJ   PIL,RNGMN1      ;DECREMENTE ADR
000250' 260740  000246*         PUSHJ   PIL,DATAR0      ;POUR LE DEUXIEME ARGUMENT - EN ABSOLU
000251' 260740  000247*         PUSHJ   PIL,RNGMN1      ;POUR LE TROISIEME ARGUMENT
000252' 262247  000000          POP     DAT,(LOC)       ;POUR LE TROISIEME ARGUMENT - EN ABSOLU
000253' 254000  000000*         JRST    RNGMEM          ;ASSURER LE RANGEMENT DU BLOC EN MEMOIRE

                                ;ROUTINE POUR PLACER EN MEMOIRE LA BONNE PAGE COMPTE TENU DES TETES
                                ;DE TAMPONS

000254' 550200  000003  IOPAGE: HRRZ    ADR,TAC2        ;PURES TETES DE TAMPONS - ICI L'ENTREE
000255' 322200  000257'         JUMPE   ADR,IOPAGF      ;SI RIEN EN ENTREE
000256' 260740  000043*         PUSHJ   PIL,ACCPAG      ;ENTRER LA BONNE PAGE
000257' 554200  000003  IOPAGF: HLRZ    ADR,TAC2        ;ICI LA SORTIE
000260' 322200  000242*         JUMPE   ADR,CPOPJ       ;SI RIEN EN SORTIE ...
000261' 306203  000000          CAIN    ADR,(TAC2)      ;MEME PAGE QUE POUR LES TAMPONS D'ENTREE ?
000262' 263740  000000          POPJ    PIL,0           ;OUI - TERMINE
000263' 606140  777777          TRNN    TAC2,-1         ;NON - EN ENTREE AUSSI ?
000264' 254000  000256*         JRST    ACCPAG          ;NON - ENTRER LA BONNE PAGE ET RETOUR
000265' 201000  000267'         MOVEI   IOPMES          ;OUI - ERREUR
000266' 254000  000200*         JRST    TAPMES          ;ENVOYER LE MESSAGE D'ERREUR

000267' 376410  143612  IOPMES: ASCIZ   .?PAGES DIFFERENTES EN E/S.
000270' 515010  444614
000271' 432132  242634
000272' 522132  320212
000273' 471010  527646
000274' 000000  000000
```

```
ENTSOR  OPERATEURS PROGRAMMES D'ENTREE-SORTIE 09.06.78  MACRO.V43 23:37  2-SEP-78 PAGE 4
ENTSOR.MAC      ROUTINE DE MANIPULATION DE FILE : UUOS LOOKUP,ENTER,RENAME

                                SUBTTL  ROUTINE DE MANIPULATION DE FILE : UUOS LOOKUP,ENTER,RENAME

000275' 541000  000000  UFILE:  HRRI    DRP,0           ;POUR RENAME PAS DE CLOSE PREALABLE
000276' 254000  000302'         JRST    UFILE0
000277' 541000  000000  UIFILE: HRRI    DRP,0           ;PREALABLE
000300' 664000  002000          TROA    DRP,ICLOSB      ;SURVEILLER L'ENTREE
000301' 541000  001000  UOFILE: HRRI    DRP,OCLOSB      ;SURVEILLER LA SORTIE
000302' 260740  000007* UFILE0: PUSHJ   PIL,UOADAB      ;ADRESSE ABSOLUE DES ARGUMENTS
000303' 541704  000000          HRRI    UUO,(ADR)       ;SAUVER L'ADRESSE ABSOLUE
000304' 260740  000600'         PUSHJ   PIL,CNVCAN      ;NUMERO DE CANAL A CHANGER
000305' 201040  000001          MOVEI   TAC,1           ;DORMIR UN PEU
000306' 047040  000031          CALLI   TAC,SLEEP
000307' 204246  000075          MOVS    DAT,JOBJDA(CHN) ;FAUT-IL UN CLOSE ?
000310' 606005  000000          TRNN    DRP,(DAT)       ;?...
000311' 260740  000345'         PUSHJ   PIL,CLOSF0      ;OUI...
000312' 201216  000000          MOVEI   ADR,(UUO)       ;ADRESSE ABSOLUE
000313' 541700  000130*         HRRI    UUO,PURARG      ;ADRESSE DE LA LISTE DES ARGUMENTS (-1 POUR PUSH)
000314' 201240  777777*         MOVEI   DAT,PURARG-1    ;....
000315' 260740  000245*         PUSHJ   PIL,ACCMEM      ;ACCES AU PREMIER ARGUMENT
000316' 261247  000000          PUSH    DAT,(LOC)       ;ET RANGEMENT EN ZONE PURE
                                                        ;POUR LES 3 ARGUMENTS SUIVANTS
                                REPEAT  3,<     PUSHJ   PIL,ACCMM1      ;ADRESSE SUIVANTE
                                        PUSH    DAT,(LOC)>      ;RANGEMENT EN ZONE PURE
000317' 260740  000230*         PUSHJ   PIL,ACCMM1      ;ADRESSE SUIVANTE
000320' 261247  000000          PUSH    DAT,(LOC)
000321' 260740  000317*         PUSHJ   PIL,ACCMM1      ;ADRESSE SUIVANTE
000322' 261247  000000          PUSH    DAT,(LOC)
000323' 260740  000321*         PUSHJ   PIL,ACCMM1      ;ADRESSE SUIVANTE
000324' 261247  000000          PUSH    DAT,(LOC)
000325' 256000  000016          XCT     UUO             ;EXECUTER L'UUO
000326' 254000  000330'         JRST    .+2             ;RETOUR D'ERREUR
000327' 350000  000020*         AOS     PAGEPC          ;RETOUR SATISFAISANT
000330' 262247  000000          POP     DAT,(LOC)       ;RANGEMENT CHEZ L'UTILISATEUR
                                                        ;POUR LES 3 ARGUMENTS QUI RESTENT
                                REPEAT 3,<      PUSHJ   PIL,RNGMM1      ;RANGER LE BLOC EN MEMOIRE S'IL FAUT EN CHANGER

                                        POP     DAT,(LOC)>      ;DEPOSER L'ARGUMENT PRECEDENT
000331' 260740  000251*         PUSHJ   PIL,RNGMM1      ;RANGER LE BLOC EN MEMOIRE S'IL FAUT EN CHANGER
000332' 262247  000000          POP     DAT,(LOC)
000333' 260740  000331*         PUSHJ   PIL,RNGMM1      ;RANGER LE BLOC EN MEMOIRE S'IL FAUT EN CHANGER
000334' 262247  000000          POP     DAT,(LOC)
000335' 260740  000333*         PUSHJ   PIL,RNGMM1      ;RANGER LE BLOC EN MEMOIRE S'IL FAUT EN CHANGER
000336' 262247  000000          POP     DAT,(LOC)
000337' 254000  000253*         JRST    RNGMEM          ;ET RANGER EN MEMOIRE CE DERNIER BLOC

                                SUBTTL  CLOSE - INPUT - OUTPUT - TAMPONS ET MODE DIRECT

000340' 260740  000600' CLSUUO: PUSHJ   PIL,CNVCAN       ;CHANGER LE NO DE CANAL
000341' 332046  000241*         SKIPE   TAC,UHHTAB(CHN) ;DES TAMPONS ?
000342' 260740  000155'         PUSHJ   PIL,IOPAG3      ;OUI - BONNE PAGE LOCALEMENT
000343' 256000  000016          XCT     UUO             ;FAIRE LE CLOSE
000344' 254000  000235'         JRST    RNGHHD          ;RENDRE LES ARGUMENTS AU PROGRAMME

000345' 332046  000341* CLOSE0: SKIPE   TAC,UHHTAB(CHN) ;DES TAMPONS ?
000346' 260740  000155'         PUSHJ   PIL,IOPAG3      ;OUI - POUR LA BONNE PAGE
000347' 135040  000670'         LDB     TAC,[POINT 3,DRP,*L<OCLOSB>]];POUR LE BON CLOSE
000350' 640040  000003          TRC     TAC,3           ;LOGIQUE DE L'INHIBITION INVERSE
000351' 505040  070000          HRLI    TAC,(CLOSE)     ;A EXECUTER
000352' 137300  000176*         DPB     CHN,PTACAC      ;COMBINER LE NO DE CANAL
000353' 256000  000001          XCT     TAC             ;EXECUTER LE CLOSE
000354' 254000  000235'         JRST    RNGHHD          ;REMETTRE EN PLACE LES ARGUMENTS EN MEMOIRE


                                ;ROUTINES POUR LES UUO'S IN ET OUT , INPUT ET OUTPUT

000355' 625000  100000  TIN:    TLZA    DRP,IO
000356' 661000  100000  TOUT:   TLO     DRP,IO
000357' 260740  000600'         PUSHJ   PIL,CNVCAN      ;CHANGER LE NO DE CANAL
000360' 336046  000345*         SKIPN   TAC,UHHTAB(CHN) ;MODE DUMP ?
000361' 254000  000413'         JRST    DUMPIO          ;OUI
000362' 205100  000400          MOVSI   TAC1,<IBUFB==400>
000363' 603000  100000          TLNE    DRP,IO          ;ENTREE ?
000364' 205100  000200          MOVSI   TAC1,<OUTBFB==200>
000365' 616106  000075          TDNN    TAC1,JOBJDA(CHN)
000366' 260740  000400'         PUSHJ   PIL,FAITMP      ;2 TAMPONS INITIAUX
000367' 260740  000201'         PUSHJ   PIL,PRDHHD      ;INSTALLATION DES TETES DE TAMPONS
000370' 260740  000254'         PUSHJ   PIL,IOPAGE      ;INSTALLATION DE LA PAGE CONCERNEE
000371' 256000  000016          XCT     UUO             ;FAIRE CE QU'IL FAUT
000372' 254000  000374'         JRST    .+2             ;RETOUR NORMAL
000373' 350000  000327*         AOS     PAGEPC          ;RETOUR AVEC SAUT
000374' 200040  000671'         MOVE    TAC,[CALLI <WAIT==10>]
000375' 137300  000352*         DPB     CHN,PTACAC      ;NUMERO DE CANAL
000376' 256000  000001          XCT     TAC             ;ATTENTE POUR SYNCHRONISATION DE I/O
000377' 254000  000235'         JRST    RNGHHD          ;RANGEMENT DES TETES DE TAMPONS
                                                        ;ET RETOUR A L'UTILISATEUR
                                ;FAIRE 2 TAMPONS AU BESOIN

000400' 261740  000016  FAITMP: PUSH    PIL,UUO         ;SAUVER L'OPERATEUR
000401' 200700  000672'         MOVE    UUO,[INBUF 2]   ;SUPPOSER ENTREE
000402' 603000  100000          TLNE    DRP,IO          ;SORTIE ?
000403' 661700  001000          TLO     UUO,(IBB)       ;OUI INBUF DEVIENT OUTBUF
000404' 137300  000000*         DPB     CHN,PUUOAC      ;NUMERO DE CANAL
000405' 260740  000035'         PUSHJ   PIL,IOBFOC      ;INBUF OU OUTBUF
000406' 262740  000016          POP     PIL,UUO
000407' 201040  000001          MOVEI   TAC,1
000410' 047040  000031          CALLI   TAC,SLEEP       ;POUR MISE A JOUR DE JOBJDA
000411' 200046  000360*         MOVE    TAC,UHHTAB(CHN)
000412' 263740  000000          POPJ    PIL,0

                                ;ENTREE-SORTIE EN MODE DUMP

000413' 625000  600000  DUMPIO: TLZA    DRP,BUFSW!SKIPSW;DRAPEAUX A-PRIORI ABAISSES
000414' 541701  000000  GOTOWD: HRRI    UUO,(TAC)       ;ADRESSE DE LA SUITE
000415' 260740  000302* DMPIO0: PUSHJ   PIL,UOADAB      ;MISE EN ABSOLU DE L'ADRESSE DE UUO
000416' 541704  000000          HRRI    UUO,(ADR)
000417' 260740  000315*         PUSHJ   PIL,ACCMEM      ;ACCES AU PREMIER TERME DE LA LISTE
000420' 336047  000000          SKIPN   TAC,(LOC)       ;DANS TAC TERME SUIVANT DE LA LISTE DE COMMANDE
000421' 254000  000545'         JRST    DMPIO5          ;0 EST LE TERMINATEUR DE LA LISTE
000422' 325040  000414'         JUMPGE  TAC,GOTOWD      ;SI > GO-TO WORD
```

```
                                ;ICI TAC CONTIENT UN VERITABLE IOWD
000423' 271040  000001          ADDI    TAC,1           ;ADRESSE DU PREMIER TERME
000424' 201201  000000          MOVEI   ADR,(TAC)       ;CETTE ADRESSE
000425' 260740  000041*         PUSHJ   PIL,METARS      ;DOIT ETRE MISE EN ABSOLU
000426' 260740  000264*         PUSHJ   PIL,ACCPAG      ;ACCES A LA PAGE CONCERNEE
000427' 261740  000001          PUSH    PIL,TAC
000430' 201257  000000          MOVEI   DAT,(PIL)
000431' 260740  000000*         PUSHJ   PIL,DATRFL      ;MISE EN RELATIF DE L'ADRESSE DE L'IOWD
000432' 262740  000001          POP     PIL,TAC
000433' 275040  000001          SUBI    TAC,1           ;REDEVIENT UN IOWD
000434' 574100  000001  DMPIO2: HLRE    TAC1,TAC        ;LONGUEUR TOTALE POUR CET IOWD
000435' 213000  000002          MOVNS   TAC1            ;-LONGUEUR
000436' 271101  000000          ADDI    TAC1,(TAC)      ;ADRESSE DU DERNIER MOT
000437' 315100  000000*         CAMGE   TAC1,PAGPLN     ;DANS LA PAGE ?
000440' 254000  000526'         JRST    DODUMP          ;OUI PAS DE PROBLEME
000441' 261740  000001          PUSH    PIL,TAC         ;SAUVER L'IOWD INITIAL
000442' 607000  200000          TLNN    DRP,BUFSW       ;DEJA CHARGE TAC2 DE LA LONGUEUR ?
000443' 260740  000114'         PUSHJ   PIL,BUFLNG      ;NON - LIRE LONGUEUR D'UN TAMPON
000444' 607000  200000          TLON    DRP,BUFSW       ;...
000445' 201141  777775          MOVEI   TAC2,-3(TAC)    ;SEULEMENT L'ENREGISTREMENT,PAS L'EN-TETE
000446' 550057  000000          HRRZ    TAC,(PIL)       ;ADRESSE DE DEPART
000447' 200100  000437*         MOVE    TAC1,PAGPLN
000450' 275100  000001          SUBI    TAC1,1          ;...
000451' 276100  000001          SUBM    TAC1,TAC        ;(TAC)=(TAC1)-(TAC)=LONGUEUR AU DESSUS
000452' 231043  000000          IDIVI   TAC,(TAC2)      ;NMB DE TAMPONS ENTIERS DANS TAC
000453' 221043  000000          IMULI   TAC,(TAC2)      ;NMB DE MOTS CORRESPONDANTS
000454' 504117  000000          HRL     TAC1,(PIL)      ;ADRESSE DE DEPART
000455' 505041  000000          HRLI    TAC,(TAC)       ;DUPLICATION
000456' 272057  000000          ADDM    TAC,(PIL)       ;MISE A JOUR DE L'IOWD DONNANT LE FUTUR IOWD
000457' 322040  000465'         JUMPE   TAC,DMPIO3      ;SAUF SI 0 (PAS DE TAMPON ENTIER DANS CETTE PAGE)

                                ;SORTIE D'AU MOINS UN TAMPON ENTIER DANS LA PAGE EN COURS
                                ;(TAC1)18-35 CONTIENT LE NMB DE MOTS RESTANT AU DESSUS DES TAMPONS ENTIERS

000460' 621040  777777          TLZ     TAC,-1          ;PARTIE DROITE =NMB DE MOTS A SORTIR
000461' 213000  000001          MOVNS   TAC             ;-NMB DE MOTS
000462' 207000  000001          MOVSS   TAC             ;EN PARTIE GAUCHE
000463' 544040  000002          HLR     TAC,TAC1        ;ADRESSE DE DEPART (IOWD D'ORIGINE)
000464' 260740  000550'         PUSHJ   PIL,XCTDMP      ;FAIRE CE DUMP PARTIEL
000465' 260740  000563' DMPIO3: PUSHJ   PIL,NULMEM      ;PRECISER LA DESTRUCTION DU BLOC DE MEMOIRE
000466' 550057  000000          HRRZ    TAC,(PIL)       ;ADRESSE (IOWD) DU RELIQUAT
000467' 622040  400000          TRZE    TAC,LOALOC      ;DE LA PAGE ?
000470' 541041  002000          HRRI    TAC,PAGLOC(TAC) ;OUI - TRANSLATER
000471' 505041  000001          HRLI    TAC,1(TAC)      ;+1 COMME ADRESSE SOURCE
000472' 541040  000000*         HRRI    TAC,MEMBLK      ;MEMBLK COMME DESTINATION
000473' 251042  777777*         BLT     TAC,MEMBLK-1(TAC1);BLT DU RELIQUAT DE LA PAGE VERS LA ZONE
                                                        ;BANALE
000474' 550200  000100*         HRRZ    ADR,PAGADR      ;ADRESSE ABSOLUE DE LA PAGE EN COURS
000475' 270200  000220*         ADD     ADR,PAGLNG      ;ADRESSE ABSOLUE DE LA SUIVANTE
000476' 260740  000426*         PUSHJ   PIL,ACCPAG      ;CHERCHER LA PAGE SUIVANTE
000477' 201042  002472*         MOVEI   TAC,MEMBLK(TAC1)        ;ADRESSE AU DESSUS DU RELIQUAT
000500' 505040  002000          HRLI    TAC,PAGLOC      ;BAS DE LA PAGE JUSTE CHARGEE
000501' 201102  000000          MOVEI   TAC1,(TAC1)     ;PARTIE DROITE SEULE
000502' 301103  000000          CAIL    TAC1,(TAC2)     ;CORRECT ? ******
000503' 254200  000503'         HALT    .               ;NON      ******
000504' 251043  777777*         BLT     TAC,MEMBLK-1(TAC2);AJOUTER LES MOTS DU BAS DE LA PAGE
                                                        ;SUIVANTE POUR FAIRE UN TAMPON PLEIN
000505' 574057  000000          HLRE    TAC,(PIL)       ;LONGUEUR RESTANT A SORTIR
000506' 213000  000001          MOVNS   TAC             ;-LONGUEUR
000507' 307043  000000          CAIG    TAC,(TAC2)      ;EST SUPERIEURE A UN TAMPON ?
000510' 254000  000531'         JRST    DMPIO4          ;NON - ALLER NE SORTIR QUE LE NECESSAIRE
                                ;ICI LE RELIQUAT=LE BAS DE LA PAGE NOUVELLE DOIVENT SORTIR COMPLETEMENT
000511' 205043  000000          MOVSI   TAC,(TAC2)      ;LONGUEUR D'UN TAMPON = CE QUI VA SORTIR
000512' 272057  000000          ADDM    TAC,(PIL)       ;MISE A JOUR DE L'IOWD DU RESTE
000513' 201043  000000          MOVEI   TAC,(TAC2)      ;LONGUEUR D'UN TAMPON
000514' 275042  000001          SUBI    TAC,+1(TAC1)    ;NMB DE MOTS AJOUTES AU RELIQUAT
000515' 271040  400000          ADDI    TAC,LOALOC      ;ADRESSE DU DEBUT DE LA SUITE A SORTIR
000516' 542057  000000          HRRM    TAC,(PIL)       ;ADRESSE POUR IOWD
000517' 211043  000000          MOVNI   TAC,(TAC2)      ;LONGUEUR DU TAMPON A SORTIR
000520' 207000  000001          MOVSS   TAC             ;COMME UN IOWD
000521' 541040  777777*         HRRI    TAC,MEMBLK-1    ;...
000522' 260740  000550'         PUSHJ   PIL,XCTDMP      ;SORTIE DU RELIQUAT+BAS DE PAGE
000523' 260740  000537'         PUSHJ   PIL,RSTRLQ      ;REPLACER EN BONNE PLACE LE DESSUS DU RELIQUAT
000524' 262740  000001          POP     PIL,TAC         ;RESTAURER L'IOWD DE PAGE COURANTE
000525' 254000  000434'         JRST    DMPIO2          ;ET POURSUIVRE AVEC CETTE PAGE

                                ;SUITE DES ENTREF-SORTIES EN MODE DUMP

000526' 260740  000550' DODUMP: PUSHJ   PIL,XCTDMP      ;EXECUTER LE DUMP TAC CONTENANT L'IOWD
000527' 350000  000016          AOS     UUO             ;ADRESSE SUIVANTE DE LA LISTE DE COMMANDE
000530' 254000  000415'         JRST    DMPIO0          ;ET CONTINUER

000531' 262740  000001  DMPIO4: POP     PIL,TAC ;RECUPERER L'IOWD DU RESTE
000532' 541040  777777*         HRRI    TAC,MEMBLK-1    ;ADRESSE DE DEPART DU RELIQUAT
000533' 260740  000550'         PUSHJ   PIL,XCTDMP      ;SORTIE D'UN TAMPON PARTIEL
000534' 260740  000537'         PUSHJ   PIL,RSTRLQ      ;RESTAURER LE RELIQUAT
000535' 350000  000016          AOS     UUO             ;ADRESSE SUIVANTE DE LA LISTE DE COMMANDE
000536' 254000  000415'         JRST    DMPIO0          ;ET CONTINUER

000537' 205042  000477* RSTRLQ: MOVSI   TAC,MEMBLK(TAC1);DEBUT DE CE QUI EST AU DESSUS DU RELIQUAT
000540' 541040  002000          HRRI    TAC,PAGLOC      ;DESTINATION
000541' 276140  000002          SUBM    TAC2,TAC1       ;TAMPON-RELIQUAT=A MOUVOIR EN PAGE NOUVELLE
000542' 271100  002000          ADDI    TAC1,PAGLOC     ;ADRESSE LIMITE
000543' 251042  777777          BLT     TAC,-1(TAC1)    ;RESTITUTION
000544' 263740  000000          POPJ    PIL,0           ;ET RETOUR

000545' 622000  400000  DMPIO5: TRZE    DRP,SKIPSW      ;EU AU MOINS UN RETOUR ERRONE ?
000546' 350000  000373*         AOS     PAGEPC          ;OUI - EN TENIR COMPTE UNE SEULE FOIS
000547' 263740  000000          POPJ    PIL,0           ;FIN DE L'UUO

000550' 622040  400000  XCTDMP: TRZE    TAC,LOALOC      ;SORTIR DE LA PAGE ?
000551' 541041  002000          HRRI    TAC,PAGLOC(TAC) ;OUI - TRANSLATER
000552' 202040  000000*         MOVEM   TAC,PUROPE      ;IOWD
000553' 402040  000001*         SETZM   TAC,PUROPE+1    ;...
000554' 261740  000016          PUSH    PIL,UUO
000555' 541700  000552*         HRRI    UUO,PUROPE      ;ADRESSE DE LA LISTE
000556' 256000  000016          XCT     UUO             ;INPUT OU OUTPUT
000557' 254000  000561'         JRST    .+2             ;RETOUR NORMAL
000560' 661000  400000          TLO     DRP,SKIPSW      ;MARQUER QU'IL Y A EU UN SKIP
000561' 262740  000016          POP     PIL,UUO         ;ORIGINAL
000562' 263740  000000          POPJ    PIL,0           ;ET RETOUR
```

```
ENTSOR  OPERATEURS PROGRAMMES D'ENTREE-SORTIE 09.08.70  MACRO.V43 23:37  2-SEP-70 PAGE 6
ENTSOR.MAC      CLOSE - INPUT - OUTPUT - TAMPONS ET MODE DIRECT

000563' 261740  000002  NULMEM: PUSH    PIL,TAC1          ;SAUVER LA LONGUEUR ISSUE DE LA PAGE D'AVANT
000564' 335100  000000*         SKIPGE  TAC1,SEGRLK       ;SEGMENT BANAL ?
000565' 260740  000000*         PUSHJ   PIL,SORSFG        ;NON - L'ECRIRE EN BONNE PLACE
000566' 402000  000000*         SETZM   PAGZDR            ;MARQUE INUTILISABLE
000567' 262740  000002          POP     PIL,TAC1          ;RECUPERATION
000570' 263740  000000          POPJ    PIL,0

000571' 260740  000600' STSSET: PUSHJ   PIL,CNVCAN         ;CONVERSION DU NO DE CANAL
000572' 201040  000017          MOVEI   TAC,17            ;POUR MASQUE
000573' 405056  000000          ANDI    TAC,(UUO)         ;MODE SEUL
000574' 301040  000015          CAIL    TAC,SD            ;CHANGEMENT EN DUMP ?
000575' 402006  000411*         SETZM   UHHTAB(CHN)       ;OUI
000576' 256000  000016          XCT     UUO               ;FAIRE LE CHANGEMENT DE MODE
000577' 263740  000000          POPJ    PIL,0             ;FIN DE L'UUO

                                SUBTTL  ROUTINES COMMUNES : CONVERSION DU CANAL,CONTROLE D'ADRESSE

                                ;ROUTINE DE CONVERSION DU NUMERO DE CANAL

000600' 201100  000017  CNVCAN: MOVEI   TAC1,17           ;4 BITS SEULEMENT
000601' 407100  000006          ANDB    TAC1,CHN          ;MASQUAGES ET RECOPIE DANS TAC1
000602' 200040  000673'         MOVE    TAC,[POINT 4,ALLCAN];ALLCAN CONTIENT L'ALLOCATION DU CANAL
000603' 133000  000001          IBP     0,TAC             ;AU MOINS UNE FOIS (MEMORY=0)
000604' 365100  000603'         SOJGE   TAC1,.-1          ;CHERCHER LA CORRESPONDANCE
000605' 135100  000001          LDB     TAC1,TAC          ;...
000606' 322100  000612'         JUMPE   TAC1,CNVCH1       ;SI NON ENCORE ATTRIBUE
000607' 200300  000002          MOVE    CHN,TAC1          ;CONVERSION DE CHN
000610' 137300  000404* CNVCH0: DPB     CHN,PUUOAC        ;PLACER DANS UUO LE NO DE CANAL
000611' 263740  000000          POPJ    PIL,0             ;ET RETOUR
000612' 201100  000001  CNVCH1: MOVEI   TAC1,1            ;POUR ARRETER 1 SECONDE (RESCHEDULE)
000613' 047100  000031          CALLI   TAC1,SLEFP        ;CE QUI ASSURE LA MISE A JOUR DE JOBJDA
000614' 332002  000075  CNVCH2: SKIPE   JOBJDA(TAC1)       ;CANAL DEJA OCCUPE ?
000615' 313100  000072          CAMLE   TAC1,JOBHCU       ;OUI - MAIS INFERIEUR AU PLUS HAUT ?
000616' 254000  000620'         JRST    CNVCH3            ;NON - PRENDRE CE PREMIER CANAL LIBRE
000617' 344100  000614'         AOJA    TAC1,CNVCH2       ;OUI - CONTINUER A CHERCHER
000620' 200300  000002  CNVCH3: MOVE    CHN,TAC1          ;VALEUR LIBRE
000621' 137300  000001          DPB     CHN,TAC           ;MARQUER LA TABLE D'ALLOCATION
000622' 307300  000017          CAIG    CHN,17            ;NON NUL ?
000623' 254000  000610'         JRST    CNVCH0            ;NON - RETOUR
000624' 201000  000626'         MOVEI   CANMES            ;SI  - MESSAGE D'ERREUR
000625' 254000  000266*         JRST    TAPMES            ;L'ENVOYER...
000626' 376411  452646  CANMES: ASCIZ   .?PLUS DE 15 CANAUX.
000627' 202110  520142
000630' 325010  340634
000631' 406533  000000

000632' 260740  000415* TTYUUO: PUSHJ   PIL,UOADAB         ;MISE EN ABSOLU DE L'ADRESSE
000633' 302300  000003          CAIE    CHN,OUTSTR        ;SORTIE D'UNE CHAINE ASCIZ ?
000634' 254000  000653'         JRST    CKADR1            ;NON ...
000635' 541700  000001          HRRI    UUO,TAC           ;OUI - ON VA LA DECOMPOSER
000636' 260740  000417* CARSU0: PUSHJ   PIL,ACCMFM        ;OBTENIR LE PREMIER MOT
000637' 205240  440707          MOVSI   DAT,(POINT 7,(LOC));FAIRE UN POINTEUR
000640' 134140  000005  CARSU1: ILDB    TAC2,DAT          ;CARACTERE SUIVANT
000641' 322140  000644'         JUMPE   TAC2,SOR5CA       ;SORTIR ET EXIT
000642' 603240  760000          TLNE    DAT,760000        ;EN BOUT DE MOT ?
000643' 254000  000640'         JRST    CARSU1            ;NON CONTINUER
000644' 200060  000005  SOR5CA: MOVE    TAC,@DAT          ;OUI - SORTIR LE MOT (AU PLUS 5 CARACTERES)
000645' 201100  000000          MOVEI   TAC1,0            ;POUR ROMPRE LA CHAINE
000646' 256000  000016          XCT     UUO               ;SORTIE EN DDT
000647' 322140  000260*         JUMPE   TAC2,CPOPJ        ;EXIT
000650' 344200  000636'         AOJA    ADR,CARSU0        ;MOT SUIVANT DE LA CHAINE
                                ;ROUTINE POUR LE GETSTS

000651' 260740  000600' STSGET: PUSHJ   PIL,CNVCAN         ;CHANGER LE NO DE CANAL
000652' 260740  000632* CKADR0: PUSHJ   PIL,UOADAB         ;CONVERSION EN ABSOLU DE L'ADRESSE DE UUO
000653' 260740  000636* CKADR1: PUSHJ   PIL,ACCMFM        ;BONNE POSITION DE L'ARGUMENT
000654' 541707  000000          HRRI    UUO,(LOC)         ;POUR FAIRE L'UUO
000655' 256000  000016          XCT     UUO               ;EXECUTER
000656' 254000  000660'         JRST    .+2               ;RETOUR SIMPLE
000657' 350000  000546*         AOS     PAGEPC            ;RETOUR AVEC SAUT
000660' 254000  000337*         JRST    RNGMEM            ;RANGER LES ARGUMENTS EN BONNE PLACE
000661'                 LIT
000661' 426351  652622
000662' 515006  020250
000663' 406332  047634
000664' 000000  000000
000665' 201000  000661'
000666' 254000  000000*
000667' 000000* 000001*
000670' 110300  000000
000671' 047000  000010
000672' 064000  000002
000673' 440400  000000*
000674'                 VAR
                        END

NO ERRORS DETECTED

PROGRAM BREAK IS 000674
```

```
ENTSOR  OPERATEURS PROGRAMMES D'ENTREE-SORTIE 09.08.70  MACRO.V43 23:37  2-SEP-70 PAGE 7
ENTSOR.MAC      SYMBOL TABLE

ACCMEM          000653' EXT     ACCMM1          000323' EXT     ACCPAG          000476' EXT     ACCUS           000273' EXT
ACU             000010  SPD     ADR             000004  SPD     ALAPRB          000003  SPD     ALLCAN          000673' EXT
ASCIIL          000001  SPD     BKSHFT  777777  777771  SPD     BLKSIZ          000200  SPD     BUFERR          000121'
BUFLNG          000114'         BUFSW           000000  SPD     CANMES          000626'         CARSU0          000636'
CARSU1          000640'         CHN             000006  SPD     CKADR0          000652'         CKADR1          000653'
CLOSE0          000345'         CLSUUO          000340' INT     CNVCAN          000600' INT     CNVCH0          000610'
CNVCH1          000612'         CNVCH2          000614'         CNVCH3          000620'         CORE            000211  SPD
CORUUO          000072' EXT     CPOPJ           000647' EXT     DAT             000005  SPD     DATAR0          000250' EXT
DATABS          000000  EXT     DATREL          000431' EXT     DATRL0          000226' EXT     DDTOUT          000273  SPD
DEVCHR          000001  SPD     DMPIO0          000415'         DMPIO2          000434'         DMPIO3          000465'
DMPIO4          000531'         DMPIO5          000545'         DODUMP          000526'         DRP             000222  SPD
DUMP            000017  SPD     DUMPIO          000413'         EXIT            000012  SPD     FAITMP          000420'
GOTOWD          000414'         HGHOFF          000002  SPD     HGHON           000003  SPD     HIGHSW  777777  777775  SPD
HORPAG          000004  SPD     IBUFB           000400  SPD     ICLOSB          000000  SPD     IDX             000211  SPD
ILEGA0          000666' EXT     ILM             020000  SPD     ILMENB          000016  SPD     IMMB            000201  SPD
INS             000016  SPD     IO              100000  SPD     IOBF0A          000043'         IOBF0B          000240'
IOBF0C          000035'         IOBUF           000032'         IOBUF0          000046'         IOBUF1          000251'
IOBUF2          000077'         IOBUF3          000103'         IOPAG1          000153'         IOPAG2          000174'
IOPAG3          000155'         IOPAGE          000254'         IOPAGF          000257'         IOPMFS          000267'
IORELE          000022' INT     JDABLK          000001  SPD     JOBFF           000045' EXT     JOBHCU          000272  INT
JOBJOA          000075  INT     JOBPAG          000140  SPD     JOBSA           000105' EXT     LOALOC          000222  SPD
LOC             000007  SPD     MEMBLK          000537' EXT     MEMORY          000000  SPD     METABS          000425' EXT
MOVHWD          000214'         MVMHWD          000244'         NADRB           000002  SPD     NORPAG          000210  SPD
NULMEM          000563'         OCLOSB          001000  SPD     OPTESS  777777  777777  SPD     OUTBFB          000200  SPD
OUTSTR          000003  SPD     PAG0LN          000447' EXT     PAGADR          000474' EXT     PAGEPC          000657' EXT
PAGHTR          000000  EXT     PAGLNG          000475' EXT     PAGLOC          000000  SPD     PAGZOR          000566' EXT
PEEK            000033  SPD     PIL             000017  SPD     PROHWD          000201'         PROHWO          000210'
PTACAC          000375' EXT     PURARG          000313' EXT     PURHWA          000201' EXT     PURHWI          000150' EXT
PURHWO          000146' EXT     PURHWZ          000203' EXT     PUROPE          000555' EXT     PURUUO          000121' EXT
PUUOAC          000610' EXT     RELLOC          000227' EXT     RLSUUO          000167' INT     RNGAR1          000132'
RNGARG          000126'         RNGHWD          000235'         RNGHWO          000241'         RNGMEM          000662' EXT
RNGMN1          000335' EXT     RSTRLO          000537'         SD              000015  SPD     SEGBLK          000564' EXT
SETDDT          000002  SPD     SKIPSW          400000  SPD     SLEEP           000031  SPD     SORSCA          000644'
SORSFG          000565' EXT     STSGET          000651' INT     STSSET          000571' INT     TAC             000221  SPD
TAC1            000002  SPD     TAC2            000003  SPD     TAPMES          000625' EXT     TIN             000355' INT
TOUT            000356' INT     TPOPJ           000233' INT     TTACAB          000137' EXT     TTY             000220  SPD
TTYUUO          000632' INT     UFILE           000275' INT     UFILE0          000302'         UHHTAB          000575' EXT
UIFILE          000277' INT     UINBF           000030' INT     UINIT           000000' INT     UOADAB          000652' EXT
UOFILE          000301' INT     UOPEN           000007' INT     UOPEN0          000011'         UOUTBF          000231' INT
USRFF           000124' EXT     USRSA           000060' EXT     UTILNG          000077' EXT     UUO             000216  SPD
UUOOFF          000001  SPD     UUOON           000000  SPD     UUOSW   777777  777775  SPD     WAIT            000210  SPD
XCTOMP          000550'         XCTUUO          000000  EXT

2K CORE USED

ACCMEM     29#   161   226   252   294   381   543   558
ACCMM1     29#   168   171   233   236   299   301   303
ACCPAG     29#   102   265   271   389   434
ACCUS      32#   123   124   126
ACU         5#
ADR         5#    57    99   101   117   124   127   128   194   197   216   217   221   222
           244   245   248   249   251   263   264   266   267   268   284   291   380   387
           432   433   553
ALAPRB      5#
ALLCAN     32#   517
ASCIIL      5#
BKSHFT      5#
BLKSIZ      5#
BUFERR     110   150#
BUFLNG     105   144#   402
BUFSW       14#   377   401   403
CANMES     535   537#
CARSU0     543#  553
CARSU1     545#  548
CHN         5#    72    83    85    97   123   124   126   144   174   177   187   191   192
           203   207   244   248   288   324   329   334   344   349   357   367   372   507
           516   522   523   531   532   533   540
CKADR0     557#
CKADR1     541   558#
CLOSE0     290   329#
CLSUUO      26   323#
CNVCAN      26    95   167   200   285   323   343   503   515#  556
CNVCH0     523#  534
CNVCH1     521   525#
CNVCH2     527#  530
CNVCH3     529   531#
CORE        5#
CORUUO      29#   125
CPOPJ       30#   222   249   267   552
DAT         5#   218   223   237   246   250   257   288   289   293   295   300   302   304
           308   314   316   318   391   544   545   547   549
DATAR0      32#   253   255
DATABS      33#
DATREL      33#   392
DATRL0      33#   227   234
DDTOUT      5#
DEVCHR      5#   145
DMPIO0     379#   465   472
DMPIO2     395#   459
DMPIO3     414   424#
DMPIO4     445   467#
DMPIO5     383   481#
DODUMP     399   463#
DRP         5#    91    92   138   140   278   280   281   282   289   331   341   342   347
           365   377   401   403   481   493
DUMP        5#
DUMPIO     345   377#
EXIT        5#
FAITMP     350   363#
GOTOWD     378#  384
HGHOFF      5#
HGHON       5#
HIGHSW      5#
HORPAG      5#
IBUFB      346#
ICLOSB      23#   281
IDX         5#
ILEGA0      33#   153
ILM         5#
ILMENB      5#
IMMB        5#     5
INS         5#
IO         15#    91    92   138   140   341   342   347   365
IOBF0A     102#   129
IOBF0B      99#   155
IOBF0C      96#   368
IOBUF       93#
IOBUF0     105#
IOBUF1     108#   114   133
IOBUF2     118    130#
```

IOBUF3	122	134#												
IOPAG1	81	187#												
IOPAG2	176	205#												
IOPAG3	189#	205	325	330										
IOPAGE	196	263#	352											
IOPAGF	264	266#												
IOPMES	272	275#												
IORELE	27	81#												
JOABLK	5#													
JOBFF	33#	104												
JOBHCU	36	37#	192	528										
JOBJOA	36	38#	144	191	288	349	527							
JOBPAG	5#													
JOBSA	33#	103	136											
LOALOC	5#	40#	230	426	451	485								
LOC	5#	103	104	162	169	172	237	257	295	300	302	304	308	314
	316	318	382	544	559									
LSTCB	5													
MEMBLK	32#	429	430	435	440	455	468	474						
MEMORY	5#													
METABS	33#	100	388											
MOVHHD	219	224#												
MVMHHD	247	251#												
NADRR	5#	5												
NORPAG	5#													
NULMEM	424	496#												
OCLOSB	24#	282	331											
OPTESS	5													
OUTBFB	348#													
OUTSTR	19#	540												
PAGØLN	30#	398	406											
PAGADR	32#	120	127	131	432									
PAGEPC	31#	57	61	62	76	307	355	482	562					
PAGHTR	30#													
PAGLNG	32#	128	228	433										
PAGLOC	5#	427	436	475	477	486								
PAGZDR	32#	499												
PEEK	5#	146												
PIL	5#	59	67	68	81	86	95	98	100	102	105	123	125	126
	148	161	167	168	171	173	176	188	193	194	195	196	197	198
	200	205	209	219	224	225	226	227	231	232	233	234	235	236
	238	239	240	247	252	253	254	255	256	265	269	283	285	290
	294	299	301	303	313	315	317	323	325	330	343	350	351	352
	363	368	369	373	379	381	388	389	390	391	392	393	400	402
	405	411	413	423	424	425	434	442	448	452	456	457	458	463
	467	469	470	479	483	489	494	495	496	498	500	501	503	509
	524	539	543	556	557	558								
PRDHHD	98	184	195	213#	351									
PRDHHO	217	220#												
PTACAC	30#	83	207	334	357									
PURARG	31#	70	73	166	170	183	292	293						
PURHHA	31#	213	214											
PURHHI	31#	141	182	218	246									
PURHHO	31#	139	180	223	250									
PURHHZ	31#	215												
PUROPE	31#	487	488	490										
PURUUO	30#	96	150											
PUUOAC	30#	367	523											
RELLOC	33#	232	235											
RLSUUO	26	200#												
RNGAR1	59	166#												
RNGARG	68	161#												
RNGHHD	77	142	244#	327	336	359								
RNGHHO	245	248#												
RNGMFM	29#	258	319	563										
RNGMN1	29#	254	256	313	315	317								
RSTRLQ	457	470	474#											
SD	5#	71	506											
SEGBLK	30#	497												
SETDDT	5#													
SKIPSW	13#	377	481	493										
SLEEP	5#	94	190	287	371	526								
SOR5CA	546	549#												
SORSEG	30#	498												
STSGET	27	556#												
STSSET	27	503#												
TAC	5#	5	58	69	70	71	82	84	93	94	97	106	107	111
	112	113	116	121	132	134	136	137	139	141	144	145	146	147
	162	166	169	170	172	174	175	177	179	181	187	202	204	206
	208	216	221	224	228	229	230	231	239	286	287	324	329	331
	332	333	335	344	356	358	370	371	372	378	382	384	386	387
	390	393	394	395	397	400	404	405	408	409	410	412	413	414
	419	420	421	422	425	426	427	428	429	430	435	436	440	442
	443	444	447	448	449	450	451	452	453	454	455	458	467	468
	474	475	478	485	486	487	488	504	505	506	517	518	528	532
	542	549												
TAC1	5#	115	117	119	120	121	130	131	132	178	180	182	183	189
	190	201	202	203	204	214	215	346	348	349	395	396	397	398
	406	407	408	411	422	430	435	437	438	450	474	476	477	478
	496	497	500	515	516	519	520	521	522	525	526	527	528	530
	531	550												
TAC2	5#	107	111	154	213	220	225	238	263	266	268	270	404	409
	410	438	440	444	447	449	453	476	545	546	552			
TAPMES	29#	273	536											
TIN	27	341#												
TOUT	27	342#												
TPOPJ	239#													
TTACAB	34#	173												
TTY	5#													
TTYUUO	27	539#												
UFILE	26	278#												
UFILEØ	279	283#												
UHHTAB	31#	72	85	97	174	177	187	203	244	248	324	329	344	372
	507													
UIFILE	26	280#												
UINBF	27	91#												
UINIT	26	57#												
UOADAB	34#	67	283	379	539	557								
UOFILE	26	282#												
UOPEN	26	67#												
UOPENØ	63	69#												
UOUTBF	27	92#												
USRFF	34#	99	101	116	134	154								
USRSA	34#	115												
UTILNG	32#	113	119	130										
UUO	5#	5	58	60	73	74	96	108	109	112	135	150	151	284
	291	292	305	326	353	363	364	366	369	378	380	464	471	489
	490	491	494	505	538	542	551	559	560					
UUOOFF	5#													
UUOON	5#													
UUOSW	5#	5												
WAIT	356#													
XCTDMP	423	456	463	469	485#									
XCTUUO	29#													